Highlights

A Lightweight 3D Anomaly Detection Method with Rotationally Invariant Features

- We propose a Point Coordinate Mapping (PCM) technique that transforms point clouds into rotation-invariant space, eliminating the adverse effects of orientation variations on feature extraction for 3D anomaly detection.

- We design a lightweight Convolutional Transform Feature Network (CTF-Net) with compositional convolution blocks to capture multi-scale structural information, achieving superior performance with only 4.1M parameters.

- We introduce Spatial 3D Data Augmentation (S3DA) for pre-training and demonstrate that our RIF framework generalizes well across different point cloud feature extractors, including PointNet, PointMLP, and Transformer-based architectures.

- Experimental results show significant improvements: 17.7% P-AUROC enhancement on Anomaly-ShapeNet and 1.6% on Real3D-AD, while maintaining computational efficiency with 0.31 FPS and 26% memory consumption.

A Lightweight 3D Anomaly Detection Method with Rotationally Invariant Features

Hanzhe Liang[a,b], Jie Zhou[c], Can Gao[*a], Bingyang Guo[d], Linlin Shen[c,e], Jinbao Wang[e,f]

[a]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, Guangdong, China*
[b]*Shenzhen Audencia Financial Technology Institute, Shenzhen University, Shenzhen, 518060, Guangdong, China*
[c]*National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen, 518060, Guangdong, China*
[d]*Software College, Northeastern University, Shenyang, 110819, Liaoning, China*
[e]*Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, 518060, Guangdong, China*
[f]*Guangdong Provincial Key Laboratory of Intelligent Information Processing, Shenzhen, 518060, Guangdong, China*

Abstract

3D anomaly detection (AD) is a crucial task in computer vision, aiming to identify anomalous points or regions from point cloud data. However, existing methods may encounter challenges when handling point clouds with changes in orientation and position because the resulting features may vary significantly. To address this problem, we propose a novel Rotationally Invariant Features (RIF) framework for 3D AD. Firstly, to remove the adverse effect of variations on point cloud data, we develop a Point Coordinate Mapping (PCM) technique, which maps each point into a rotationally invariant space to maintain consistency of representation. Then, to learn robust and discriminative features, we design a lightweight Convolutional Transform Feature Network (CTF-Net) to extract rotationally invariant features for the memory bank. To improve the ability of the feature extractor, we introduce the idea of transfer learning to pre-train the feature extractor with 3D data augmentation. Experimental results show that the proposed method achieves the advanced performance on the Anomaly-ShapeNet dataset, with an average P-AUROC improvement of 17.7%, and also gains the best performance on the Real3D-AD dataset, with an average P-AUROC improvement of 1.6%. The strong generalization ability of RIF has been verified by combining it with traditional feature extraction methods on anomaly detection tasks, demonstrating great potential for industrial applications. The code is available at `https://github.com/hzzzzzhappy/RIF`.

Keywords: Anomaly detection, point cloud, rotationally invariant, convolution.

1. Introduction

3D anomaly detection (AD) [3, 4, 5, 6] has attracted increasing attention from the computer vision community due to its widespread application in inspecting high-precision industrial products. It is dedicated to identifying anomalous points or regions that deviate from given 3D point cloud data. In real-world industrial applications, we often face a scenario where it is easy to collect normal samples but time-consuming and costly to acquire defective samples, such that unsupervised methods that only learn from normal samples are widely used.

For 3D AD, some traditional methods such as BTF [7] have been proposed to extract features directly from 3D structures, but they have limitations in extracting cross-sample features to accommodate the diversity of different samples. In view of the great success of deep learning, deep neural network-based methods have become the mainstream approach for 3D AD. These methods can be roughly categorized into feature embedding-based [1] and reconstruction-based [8].
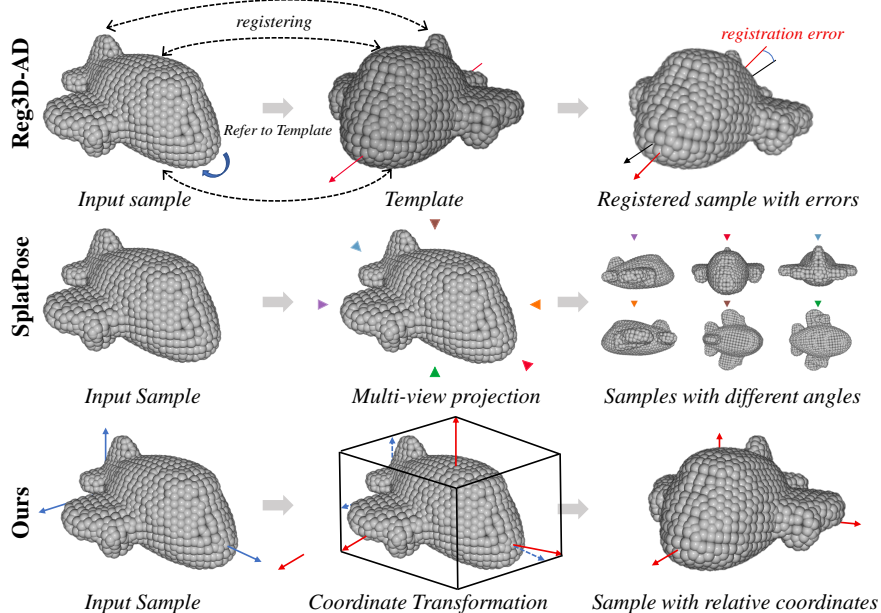
Figure 1: Comparison of representative methods for point clouds with variations in orientation and position. Reg3D-AD [1] aligns the testing sample with the standard template by registration. SplatPose [2] reconstructs 3D object from multi-view images by Gaussian splash. Our method maps point clouds into a rotational invariance space by coordinate transformation.

Firstly, feature embedding-based methods use pre-trained models to extract features of normal samples for a memory bank, with anomalies detected by comparing test sample features to those in the bank. Some methods such as M3DM [9] have achieved on the datasets like MvTec3D-AD [10] using memory bank methods. However, they face challenges on unregistered datasets such as Real3D-AD [1] and Anomaly-ShapeNet [4]. Reg3D-AD [1] used PointMAE [11] for feature representation based on the registered point clouds, and Group3AD [3] further used contrast learning to improve feature differentiation ability. Secondly, reconstruction-based methods encode point cloud data into a latent space and decode it in its original form, with high reconstruction errors indicating anomalies. IMRNet [4] improved anomaly detection by using geometric-preserving downsampling and random masking. R3D-AD [8] utilized PointNet [12] and a diffusion process to progressively restore point cloud. These methods have shown promising results on the Anomaly-ShapeNet [4] and Real3D-AD [1] datasets.

Industrial 3D AD is particularly difficult because products on the production line may appear in different poses. Consequently, the same product may have point cloud data with variations in orientation and position. Although a rotating point cloud does not alter its geometric structure, it changes the coordinate representation corresponding to the structure, resulting in a significant difference in the output of the feature extraction network. Reg3D-AD [1] aligns test samples with a standard template via registration to reduce data variation effects, but this method is computationally intensive and risks registration errors, particularly with significant train-test gaps. SplatPose [2] reconstructs 3D objects via Gaussian splatting to enhance pose robustness; however, its point cloud precision relies on image quality and view count, and it requires test image-to-3D alignment. This inspires us to find a unique representation of point cloud data to maintain rotation invariance. Therefore, our method proposed to map point clouds into a rotational invariance space through coordinate transformation, enabling the generation of a unique feature representation for point clouds with different rotations and translations. The differences between our solution and previous methods

are shown in Figure 1.

Motivated by the above facts, we present the Rotationally Invariant Features (RIF) framework. Specifically, we first introduce a Point Coordinate Mapping (PCM) technique to transform point cloud coordinates into the relative coordinate system. Then, we design a Convolutional Transform Feature Network (CTF-Net) based on Compositional Convolution, capturing local point cloud features at different scales. To assist the pre-training of CTF-Net, we develop a Spatial 3D Data Augmentation (S3DA) technique to diversify training samples for better representation. Through comparative experiments and ablation studies on Anomaly-ShapeNet and Real3D-AD, it is observed that our framework is highly appealing and achieves state-of-the-art results in various metrics. To the best of our knowledge, we are the first method to employ a deep feature extractor for the rotationally invariant feature representation of point clouds. The contributions are summarized as follows:

- To eliminate the adverse effects of point cloud data variations on feature extraction, we develop a point coordinate mapping (PCM) technique, which maps point clouds to a rotation-invariant space, enabling consistent point coordinate representation for downstream feature extraction.

- To effectively extract point cloud features, we design a Convolutional Transform Feature Network (CTF-Net), which performs multiple 1D convolutions at different scales to capture local structure information. Additionally, we introduce the Spatial 3D Data Augmentation (S3DA) strategy to aid the pre-training of our feature extraction network, providing diverse training data to enhance feature representation.

- Based on PCM and CTF-Net, we propose the Rotationally Invariant Features (RIF) framework for anomaly detection. Our framework can seamlessly integrate with other point cloud feature extractors, providing a general and practical solution for 3D anomaly detection.

- We conducted extensive comparative and ablation experiments on the Anomaly-ShapeNet and Real3D-AD datasets. Experimental results show the superiority of our method, with a P-AUROC improvement of 17.7% on Anomaly-ShapeNet and 1.6% on Real3D-AD.

The rest of this paper is organized as follows. In Section 2, we discuss related work. In Section 3, we present our method. In Section 4, we validate the proposed method through detailed experiments. In Section 6.1, we provide a theoretical analysis of the proposed method. Main symbols in the paper and their explanations are reported in Table 1 to help understand the paper.

2. Related Work

2.1. 2D Anomaly Detection

2D anomaly detection [13, 14, 15] is an important vision task that aims to detect and localize anomalies from images. Existing 2D AD methods are mainly classified into two categories: generative and discriminative methods [16]. Generative models such as Autocoder [17], GAN [18], and Diffusion [19] are employed to learn the feature representation or distribution of normal samples, and anomalies are detected by performing image reconstruction [20] or matching with memory bank [21, 22]. Recent methods, such as DRAEM [23], PatchCore [21] and CRAD [24] have yielded impressive results. Discriminative methods aim to train a supervised model to discriminate anomalous and normal images, which usually requires a certain number of anomalous samples annotated by humans or generated by data augmentation. Some methods like DRA [25], BGAD [26], and AHL [27] have shown the benefits of supervised information to improve 2D anomaly detection performance. Although promising results, applying 2D anomaly detection methods to 3D point clouds directly may face significant challenges.

Notation	Description
$\mathbf{P}_{(n,3)}$	Raw point cloud containing n 3D points
$\mathcal{F}: \mathbf{P}_{(n,3)} \rightarrow \mathbf{F}_{(n',d)}$	Feature extractor mapping a point cloud to d-dimensional features
$\mathbf{c} \in \mathbb{R}^3$	The centroid of a given point cloud: $\mathbf{c} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_i$
$\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3 \in \mathbb{R}^3$	Basis vectors: $\mathbf{u}_1 = \mathbf{p}_{\text{far1}} - \mathbf{c}$, $\mathbf{u}_2 = \mathbf{p}_{\text{far2}} - \mathbf{c}$, $\mathbf{u}_3 = \mathbf{p}_{\text{near}} - \mathbf{c}$
$\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathbb{R}^3$	Orthonormal basis via Gram-Schmidt orthogonalization
$\mathbf{p}_i^* \in \mathbb{R}^3$	Mapped point coordinates: $\mathbf{p}_i^* = (\mathbf{p}_i - \mathbf{c})[\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3]^T$
$S(\cdot), J(\cdot), Z(\cdot)$	S3DA operations: anisotropic scaling, jittering, zero-masking
$\mathbf{V}_{(1,512)} \in \mathbb{R}^{512}$	Global feature vector from CTF-Net
$\mathbf{M}_{3\times3} \in \mathbb{R}^{3\times3}$	Transformation matrix generated by MLP
$\mathbf{F}_{(1,1024)}^* \in \mathbb{R}^{1024}$	Final sample-level feature vector
$G \in \mathbb{N}^+$	Number of groups (default: 512)
$K \in \mathbb{N}^+$	Number of points per group (default: 512)
Mem	Memory bank storing normal features
$\|\cdot\|_2$	L_2-norm for nearest neighbor distance
P-AUROC	Pixel-level Area Under ROC Curve
O-AUROC	Object-level Area Under ROC Curve
P-AUPRO	Pixel-level Area Under PRO Curve
O-AUPRO	Object-level Area Under PRO Curve

Table 1: Main symbols and their explanations in the paper.

2.2. 3D Anomaly Detection

3D anomaly detection intends to identify and locate anomalous points or regions within 3D point cloud data. Existing methods for 3D AD can be grouped into two main categories: feature embedding-based methods and reconstruction-based methods. Feature embedding-based methods use pre-trained models to extract normal features for a memory bank, detecting anomalies by comparing test features against stored entries. Reg3D-AD [1] uses PointMAE [11] to extract global geometric and local coordinate features post-registration, matching both during inference for anomaly scoring. Group3AD [3] clusters groups into uniformly compact structures, constructing group-level features for the memory bank

Reconstruction-based methods aim to project point cloud data into a high-dimensional latent space through encoding processes, subsequently reconstructing the original input via decoding, with points demonstrating elevated reconstruction errors being identified as anomalies.IMRNet [4, 28] improved PointMAE by geometric-preserving downsampling and random masking to enhance reconstruction ability for anomaly detection. R3DAD [8] employed PointNet to progressively restore full-mask point clouds by performing the diffusion, aiding the model in accurately identifying abnormal regions. Additionally, some methods integrate 2D images with 3D point clouds to perform RGB-D AD. M3DM [9] aligns PointTransformer (3D) and ViT (2D) features via contrastive learning. CPMF [29] combines local 3D geometry with 2D semantics from multi-view images. Looking3D [30] integrates texture-enhanced 3D data with cross-modality matching.

In 3D point cloud AD [31], data variations in orientation significantly affect detection performance. IMRNet [4] and R3DAD [8] used data augmentation techniques to introduce diversity through performing geometric transformations, noise injection, and random masking on point clouds. Reg3D-AD [1] and Group3AD [3] introduced registration to align different point cloud samples to
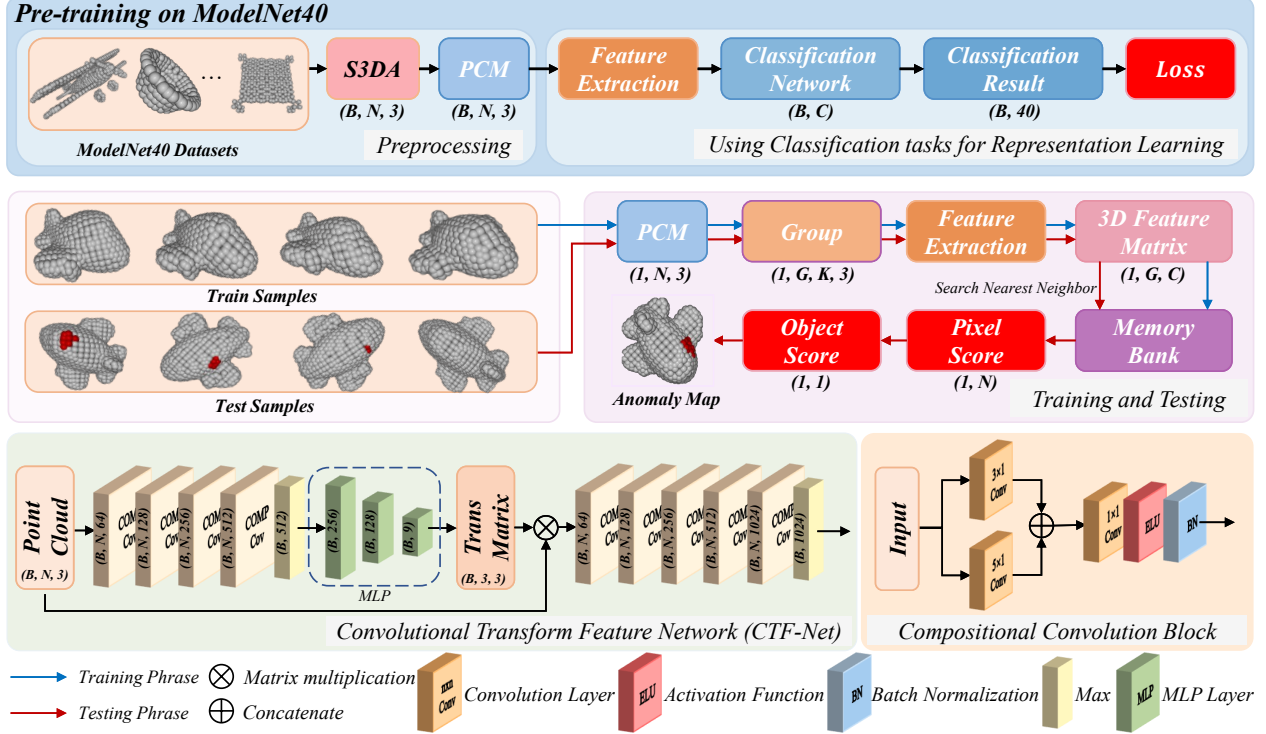
Figure 2: Pipeline of our RIF framework. Feature extractor CTF-Net is first pre-trained on the ModelNet40 dataset with 3D data augmentation and coordinate mapping. Then, normal training samples from the target dataset are mapped and grouped for the pre-trained CTF-Net to extract discriminative features for a memory bank. Finally, the extracted features of testing samples are matched with the memory bank to generate pixel and object scores by nearest neighbor searching.

improve feature extraction. CPMF [29], Looking3D [30], and SplatPose [2] employed multi-view strategies to capture features from different angles to enhance the robustness of anomaly detection. Although these methods partially alleviate the adverse effect of orientation on detection performance, there is still a strong desire to develop a simple and computationally efficient technique to accommodate 3D point cloud data with arbitrary orientation.

3. Approach

The core of our RIF framework is to learn rotationally invariant features to address the diversity of samples in orientation, position, and posture. Here, we propose a simple yet efficient solution for 3D AD, which is shown in Figure 2. We will elaborate on its details in the following sections.

3.1. Point Coordinate Mapping for Rotational Invariance

For 3D point cloud data, maintaining rotational invariance is crucial for ensuring model robustness. Existing feature extractors try to learn representative features from each sample, i.e., $\mathcal{F} : \mathbf{P}_{(n,3)} \rightarrow \mathbf{F}_{(n',d)}$, where $\mathbf{P}_{(n,3)}$ denotes the set of 3D points with the number of n, and $\mathbf{F}_{(n',d)}$ represents the learned feature map with the size of $n' * d$. However, in this process, any rotation imposed on the input point cloud may alter the output features, affecting the robustness of features and performance of the trained model. To address this problem, we propose a PCM module to transform point cloud samples. Specifically, the PCM aims to construct a relative coordinate representation for each sample with any rotation and translation. It calculates the centroid $\mathbf{c}$ of each

sample: $\mathbf{c} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{p}_i$, where $\mathbf{p}_i$ is a point within the sample, and n is the number of all points. Then, PCM sequentially selects three key vectors to define the relative coordinate system in the following manner:

- $\mathbf{u}_1 = \mathbf{p}_{\text{far1}} - \mathbf{c}$, where $\mathbf{p}_{\text{far1}}$ is the farthest point from the centroid;

- $\mathbf{u}_2 = \mathbf{p}_{\text{far2}} - \mathbf{c}$, where $\mathbf{p}_{\text{far2}}$ is the second farthest point from the centroid, and is linearly independent to $\mathbf{u}_1$;

- $\mathbf{u}_3 = \mathbf{p}_{\text{near}} - \mathbf{c}$, where $\mathbf{p}_{\text{near}}$ is the nearest point to the centroid that makes the rank of the three selected vectors equal to 3, i.e., $Rank([\mathbf{u}_1; \mathbf{u}_2; \mathbf{u}_3]) = 3$.

It is well-known that spanning a 3D space requires at least three vectors. In case the selected vectors are linearly dependent, PCM sequentially selects the next farthest or nearest points until the constraints are satisfied. After determining these three vectors, Schmidt orthogonalization is performed to generate standard orthogonal basis vectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$. For each point, it can be transformed into the relative coordinate point by:

$$\mathbf{p}_i^* = (\mathbf{p}_i - \mathbf{c})[\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3]^{\mathrm{T}}. \tag{1}$$

Following the Eq. (16), a point cloud sample with any rotation and translation can be mapped into a relative coordinate representation by the PCM: $\mathcal{M} : \mathbf{P}_{(n,3)} \rightarrow \mathbf{P}_{(n,3)}^*$ (See *Supplementary Materials* for the theoretical proof). This mapping ensures that each sample with different rotations and translations has the same coordinate representation, providing a stable input for downstream feature extraction.

3.2. Pre-trained Feature Extractor with Spatial 3D Data Augmentation

Due to the limited number of training samples in 3D anomaly detection, learning a good feature extractor may face great challenges. To enhance the feature extraction ability for 3D point cloud data, we resort to the technique of network pre-training and transferring. Specifically, the feature extractor is pre-trained from scratch on the ModelNet40 dataset. To enhance data diversity and feature generalization, we introduce Spatial 3D Data Augmentation (S3DA) as a complementary approach to PCM. It diversifies training data by applying additional augmentation techniques to introduce spatial variability without compromising consistency, while PCM realizes rotational invariance through mapping points into relative coordinate representation. S3DA achieves this by combining several augmentation strategies, including random scaling along different axes, point-level random jittering, and random zeroing of specific points. These operations can be represented as:

$$\mathbf{P}_{(n,3)}' = Z(J(S(\mathbf{P}_{(n,3)}))), \tag{2}$$

where $\mathbf{P}_{(n,3)}$ and $\mathbf{P}_{(n,3)}'$ denote the original and augmented pointclouds, respectively, $S(\cdot)$, $J(\cdot)$, and $Z(\cdot)$ mean the random scaling, jittering, and zeroing operations, respectively.

After augmentation and mapping, the preprocessed ModelNet40 [32] dataset is used to pre-train the feature extractor to learn the generalized representation ability on the classification task. By pre-training on the preprocessed 3D point cloud data, the feature extractor is exposed to a wider range of spatial variations in training, which not only promotes the learning of more versatile and discriminative feature representation but also enhances the model's robustness in downstream tasks.

3.3. Convolutional Transform Feature Network for Anomaly Detection

By utilizing the proxy classification task, the feature extractor has a versatile representation ability for point clouds. The ability to capture multi-scale structural information in a lightweight manner is crucial for achieving high detection performance. To address this problem, we propose a lightweight CTF-Net for extracting multi-scale features in 3D anomaly detection. As shown in Figure 2, it consists of several compositional convolution blocks (CCB) and Multilayer Perceptron (MLP). Specifically, each CCB performs the 1D convolutions with the kernel sizes of $3*1$ and $5*1$ on the input, respectively, and then concatenates the two convolution results to capture local information of each point at different scales. Besides, the 1D convolution with the kernel size of $1*1$ is applied to the concatenated features, followed by the Exponential Linear Units (ELU) activation function and batch normalization (BN).

For any point cloud sample, CTF-Net uses four CCBs to enhance the dimensionality of each point to 512. Then the max operation is imposed on the feature map in the dimensionality of the sample size, and each point cloud sample is represented into a vector with a size of 512. Formally, this process can be described as:

$$\mathbf{V}_{(1,512)} = \max\{\text{CCB}_4(\text{CCB}_3(\text{CCB}_2(\text{CCB}_1(\mathbf{P}_{(n,3)}))))\}. \tag{3}$$

After that, the feature vector $\mathbf{V}$ is mapped to a transformation matrix $\mathbf{M}$ with a size of 3*3 by MLP, and the original point cloud $\mathbf{P}_{(n,3)}$ is multiplied with this transformation matrix $\mathbf{P}_{(3,3)}$ to obtain a better representation $\mathbf{F}_{(n,3)}$ of all points, which can be formally expressed as:

$$\mathbf{F}_{(n,3)} = \mathbf{P}_{(n,3)} \otimes MLP(\mathbf{V}_{(1,512)}). \tag{4}$$

Subsequently, each transformed point is expanded to the dimensionality of 1024 by using multiple CCB, and each point cloud sample is mapped to a final feature vector with the dimensionality of 1024, which can be formulated as:

$$\mathbf{F}^*_{(1,1024)} = \max\{\text{CCB}_5(\cdots(\text{CCB}_1(\mathbf{F}_{(n,3)})))\}. \tag{5}$$

With the carefully designed feature extraction network CTF-Net, we can train a model for 3D anomaly detection. During the training phase, the input point cloud data is first performed coordinate mapping by using the PCM module, and all points are divided into several groups through farthest point sampling. This process can be described as:

$$\mathbf{p}_i = FPS(PCM(\mathbf{P}_{(n,3)})), \tag{6}$$

$$\mathbf{G}_i = KNN(\mathbf{p}_i), \tag{7}$$

where $\mathbf{p}_i$ denotes the selected point by Farthest Point Sampling (FPS) from point clouds after PCM coordinate mapping, $\mathbf{G}_i$ means the i-th group through searching k-nearest neighbors (KNN) for the point $\mathbf{p}_i$.

For each group with k points, the CTF-Net extracts features separately and the resulting features are stored in a memory bank:

$$\mathbf{Mem}_i = CTF\text{-}Net(\mathbf{G}_i). \tag{8}$$

Through this feature extraction process, we can learn the distribution of group features for normal samples to construct a memory bank for anomaly detection.

During the testing phase, the testing sample is similarly divided into several groups as in the training phase, and feature extraction is performed on each group separately. For each extracted

group feature, its distance to the nearest neighbor in the memory bank is considered as the score, which can be calculated as:

$$\mathbf{S}_i = \| \mathbf{Mem}_{near} - \mathbf{F}_i \|, \tag{9}$$

where $\mathbf{F}_i$ and $\mathbf{Mem}_{near}$ denote the extracted group feature and its nearest neighbor feature in the memory bank, respectively, and the symbol $\| \cdot \|$ represents the 2-norm. This score can be used as the pixel-level score to reflect the degree of abnormality of the selected point $\mathbf{p}_i$ by FPS. While the maximum value of all point scores can be used to define the object-level score of the testing sample.

3.4. Pseudo Code

We provide pseudo-code for PCM in Algorithm 1 to help better understand the data flow of our model. All source codes will be released upon acceptance.

Algorithm 1 Point Cloud Mapping (PCM)

Require: Point cloud $\mathbf{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^n$
Ensure: Rotation-invariant representation $\mathbf{P}^* = \{\mathbf{p}_i^* \in \mathbb{R}^3\}_{i=1}^n$
 1: **Step 1: Center the point cloud**
 2: Compute centroid: $\mathbf{c} \leftarrow \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i$
 3: Center points: $\widetilde{\mathbf{p}}_i \leftarrow \mathbf{p}_i - \mathbf{c}, \; \forall i \in \{1, \ldots, n\}$
 4: **Step 2: Sort points by distance from origin**
 5: Sort indices $I \leftarrow \mathrm{argsort}(\|\widetilde{\mathbf{p}}_i\|_2, \; \text{descending})$
 6: $\mathbf{u}_1 \leftarrow \widetilde{\mathbf{p}}_{I[1]}$ ▷ First basis vector candidate
 7: **Step 3: Find linearly independent second vector**
 8: **for** $k \leftarrow 2$ to n **do**
 9:     **if** $\mathrm{rank}\left([\mathbf{u}_1, \widetilde{\mathbf{p}}_{I[k]}]\right) = 2$ **then**
10:         $i_2 \leftarrow I[k], \; \mathbf{u}_2 \leftarrow \widetilde{\mathbf{p}}_{i_2}$
11:         **break**
12:     **end if**
13: **end for**
14: **Step 4: Find linearly independent third vector**
15: **for** $k \leftarrow n$ down to 1 **do**
16:     $\mathbf{U} \leftarrow [\mathbf{u}_1, \mathbf{u}_2, \widetilde{\mathbf{p}}_{I[k]}]$
17:     **if** $\mathrm{rank}(\mathbf{U}) = 3$ **then**
18:         $i_3 \leftarrow I[k], \; \mathbf{u}_3 \leftarrow \widetilde{\mathbf{p}}_{i_3}$
19:         **break**
20:     **end if**
21: **end for**
22: **Step 5: Gram-Schmidt orthogonalization**
23: Compute basis vector $\mathbf{e}_1 \leftarrow \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|_2}$
24: $\mathbf{v}_2 \leftarrow \mathbf{u}_2 - (\mathbf{u}_2^\top \mathbf{e}_1)\mathbf{e}_1$
25: Compute basis vector $\mathbf{e}_2 \leftarrow \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|_2}$
26: $\mathbf{v}_3 \leftarrow \mathbf{u}_3 - \sum_{j=1}^2 (\mathbf{u}_3^\top \mathbf{e}_j)\mathbf{e}_j$
27: Compute basis vector $\mathbf{e}_3 \leftarrow \frac{\mathbf{v}_3}{\|\mathbf{v}_3\|_2}$
28: **Step 6: Apply transformation**
29: Compute transformation matrix $\mathbf{S} \leftarrow [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] \in \mathbb{R}^{3 \times 3}$
30: Transform points: $\mathbf{p}_i^* \leftarrow \mathbf{S}^\top (\mathbf{p}_i - \mathbf{c}), \; \forall i \in \{1, \ldots, n\}$
31: **return** $\mathbf{P}^* = \{\mathbf{p}_i^*\}_{i=1}^n$

---

4. Experiments

In this section, comparative experiments are first conducted to show the effectiveness of our method, followed by ablation studies and parameter sensitivity analysis to examine the rationality of the key components. Finally, extensibility experiments are also performed to demonstrate the generalization of our framework RIF with other feature extractors.

Method	cap0	cap3	helmet3	cup0	bowl4	vase3	headset1	eraser0	vase8	cap4	vase2	vase4	helmet0	bucket1
BTF(Raw)	0.524	0.687	0.700	0.632	0.563	0.602	0.475	0.637	0.550	0.469	0.403	0.613	0.504	0.686
BTF(FPFH)	0.730	0.658	0.724	0.790	0.679	0.699	0.591	0.719	0.662	0.524	0.646	0.710	0.575	0.633
M3DM	0.531	0.605	0.655	0.715	0.624	0.658	0.585	0.710	0.551	0.718	0.737	0.655	0.599	0.699
PatchCore(FPFH)	0.472	0.653	0.737	0.655	0.720	0.430	0.464	0.810	0.575	0.595	0.721	0.505	0.548	0.571
PatchCore(PointMAE)	0.544	0.488	0.615	0.510	0.501	0.465	0.423	0.378	0.364	0.725	0.742	0.523	0.580	0.574
RegAD	0.632	0.718	0.620	0.685	0.800	0.511	0.626	0.755	0.811	0.815	0.405	0.755	0.600	0.752
IMRNet	0.715	0.706	0.663	0.643	0.576	0.401	0.476	0.548	0.635	0.753	0.614	0.524	0.598	0.774
CPMF	0.601	0.551	0.520	0.497	0.683	0.582	0.458	0.689	0.529	0.553	0.582	0.514	0.555	0.601
R3D-AD	0.666	0.676	0.708	0.600	0.651	0.595	0.676	0.660	0.597	0.678	0.712	0.674	0.623	0.615
Ours	**0.968**	**0.955**	0.719	**0.912**	**0.802**	**0.863**	**0.840**	**0.976**	**0.938**	**0.956**	**0.930**	**0.842**	**0.838**	**0.812**

Method	bottle3	vase0	bottle0	tap1	bowl0	bucket0	vase5	vase1	vase9	ashtray0	bottle1	tap0	phone	cup1
BTF(Raw)	0.720	0.618	0.551	0.564	0.524	0.617	0.585	0.549	0.564	0.512	0.491	0.527	0.583	0.561
BTF(FPFH)	0.622	0.642	0.641	0.596	0.710	0.401	0.429	0.619	0.568	0.624	0.549	0.568	0.675	0.619
M3DM	0.532	0.608	0.663	0.712	0.658	**0.698**	0.642	0.602	0.663	0.577	0.637	0.654	0.358	0.556
PatchCore(FPFH)	0.512	0.655	0.654	0.768	0.524	0.459	0.447	0.453	0.663	0.597	0.687	0.733	0.488	0.596
PatchCore(PointMAE)	0.653	0.677	0.553	0.541	0.527	0.586	0.572	0.551	0.423	0.495	0.606	**0.858**	0.886	0.856
RegAD	0.525	0.548	0.886	**0.741**	0.775	0.619	0.624	0.602	0.694	0.698	0.696	0.589	0.599	0.698
IMRNet	0.641	0.535	0.556	0.699	0.781	0.585	0.682	0.685	0.691	0.671	0.702	0.681	0.742	0.688
CPMF	0.435	0.458	0.521	0.657	0.745	0.486	0.651	0.486	0.545	0.615	0.571	0.458	0.545	0.509
R3D-AD	0.750	0.743	0.722	0.716	0.711	0.593	0.642	0.616	0.617	0.689	0.674	0.643	0.693	0.617
Ours	**0.813**	**0.905**	**0.952**	0.667	**0.979**	0.553	**0.696**	**0.807**	**0.775**	**0.865**	**0.710**	0.670	**0.955**	0.831

Method	vase7	helmet2	cap5	shelf0	bowl5	bowl3	helmet1	bowl1	headset0	bag0	bowl2	jar	Mean
BTF(Raw)	0.578	0.605	0.373	0.464	0.517	0.685	0.449	0.464	0.578	0.430	0.426	0.423	0.550
BTF(FPFH)	0.540	0.643	0.586	0.619	0.699	0.590	**0.749**	**0.768**	0.620	0.746	0.518	0.427	0.628
M3DM	0.517	0.623	0.655	0.554	0.489	0.657	0.427	0.663	0.581	0.637	0.694	0.541	0.616
PatchCore(FPFH)	0.693	0.455	0.795	0.613	0.358	0.327	0.489	0.531	0.583	0.574	0.625	0.478	0.580
PatchCore(PointMAE)	0.651	0.651	0.545	0.543	0.562	0.581	0.562	0.524	0.575	0.674	0.515	0.487	0.577
RegAD	0.881	0.825	0.467	0.688	0.691	0.654	0.624	0.615	0.580	0.715	0.593	0.599	0.668
IMRNet	0.593	0.644	0.742	0.605	0.715	0.599	0.604	0.705	0.705	0.668	0.684	0.765	0.650
CPMF	0.504	0.515	0.551	**0.783**	0.684	0.641	0.542	0.488	0.699	0.655	0.635	0.611	0.573
R3D-AD	0.619	0.598	0.662	0.681	0.572	0.633	0.651	0.742	0.637	0.654	0.550	0.736	0.657
Ours	**0.913**	**0.905**	**0.943**	0.765	**0.927**	**0.872**	0.639	0.613	**0.895**	**0.957**	**0.859**	**0.967**	**0.845**

Table 2: P-AUROC performance of different methods on Anomaly-ShapeNet across 40 categories, where the best and second-place results are highlighted in **bold** and <u>underline</u>, respectively.

4.1. Experimental Setup

Datasets. We conducted comparative experiments on both available real or synthetic 3D AD datasets, namely Anomaly-ShapeNet and Real3D-AD. Anomaly-ShapeNet has over 1,600 samples from 40 categories. The training set contains only four normal samples for each category, while the test set includes both normal and anomalous samples with diverse defects. Real3D-AD is a high-resolution 3D dataset, containing 1,254 samples from 12 categories. The training set for each category includes four normal samples, while the test set for each category contains both normal and anomalous samples with different defects.

Baselines. Our method was compared with eight recently SOTA methods, including BTF [7], M3DM [9], PatchCore [21], CPMF [29], R3D-AD [8], Group3AD [3], Reg3D-AD [1], MC3D-AD [33] and IMRNet [4]. The results of these methods are excerpted from their papers or implemented by publicly available codes.

Evaluation Metrics. We used P-AUROC (Area Under the Receiver Operator Curve↑) to evaluate pixel-level anomaly localisation accuracy and O-AUROC(↑) to assess object-level AD capability. Additionally, we adopted the metrics of P-AUPRO (Area Under the Per-Region-Overlap↑) and O-AUPRO(↑) to evaluate pixel-level and object-level performance, respectively. Moreover, we employed Frames Per Second (FPS↑) and Memory (↓) to evaluate the inference speed and resource consumption.

Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Mean
BTF(Raw)	0.564	0.647	0.735	0.608	0.563	0.601	0.514	0.597	0.520	0.489	0.392	0.623	0.571
BTF(FPFH)	**0.738**	0.708	0.864	0.693	0.882	**0.875**	0.709	<u>0.891</u>	0.512	0.571	0.501	0.815	0.730
M3DM(PointBERT)	0.523	0.593	0.682	**0.790**	0.594	0.668	0.589	0.646	0.574	0.732	0.563	0.677	0.636
M3DM(PointMAE)	0.530	0.607	0.683	0.735	0.618	0.678	0.600	0.654	0.561	0.748	0.555	0.679	0.637
PatchCore(FPFH)	0.471	0.643	0.637	0.618	0.760	0.430	0.464	0.830	0.544	0.596	0.522	0.411	0.577
PatchCore(FPFH+Raw)	0.556	0.740	0.749	0.558	0.854	0.658	0.781	0.539	0.808	0.753	0.613	0.549	0.680
PatchCore(PointMAE)	0.579	0.610	0.635	0.683	0.776	0.439	0.714	0.514	0.660	0.725	0.641	0.727	0.642
Reg3D-AD	0.631	0.718	0.724	0.676	0.835	0.503	0.826	0.545	0.817	0.811	0.617	0.759	0.705
R3D-AD	0.594	0.557	0.593	0.620	0.555	0.635	0.668	0.562	0.578	0.578	0.608	0.568	0.592
CPMF	0.618	**0.836**	0.734	0.559	0.753	0.719	**0.988**	0.449	**0.962**	0.725	**0.800**	**0.959**	0.758
Group3AD	0.636	0.745	0.738	<u>0.759</u>	0.862	0.631	0.836	0.564	<u>0.827</u>	**0.798**	0.625	0.803	0.735
MC3D-AD	0.628	<u>0.819</u>	<u>0.910</u>	0.640	<u>0.942</u>	0.822	<u>0.932</u>	0.458	0.659	<u>0.778</u>	<u>0.690</u>	<u>0.934</u>	<u>0.768</u>
Ours	<u>0.733</u>	0.777	**0.913**	0.728	**0.975**	<u>0.869</u>	0.710	**0.935**	0.573	0.714	0.588	0.888	**0.784**

Table 3: P-AUROC performance of different methods on Real3D-AD across 12 categories, where the best and the second-place results are highlighted in **bold** and <u>underline</u>, respectively.

Implementation Details. The proposed feature extractor CTF-Net was pre-trained from scratch on the ModelNet40 dataset, with each sample downsampling from 1024 to 512 points to match downstream tasks. During training and testing, the group size was set to 512 for feature extraction, with 512 points in each group. All codes were implemented in PyTorch using Python 3.8, and the experiments were conducted on a server equipped with an RTX 3090 (24GB).

4.2. Experimental Results and Analysis

To show the validity of our method, we conducted comparison experiments on Anomaly-ShapeNet and Real3D-AD, and the performance in terms of P-AUROC is shown in Tables 2 and 3, respectively. Prior work has demonstrated that rotation-invariant representation methods often exhibit diminished performance on general downstream tasks [34, 35]. In contrast, our proposed approach successfully reconciles this trade-off by achieving state-of-the-art AD performance while preserving robust rotation-invariant properties.

1) Comparison on Anomaly-ShapeNet. As shown in Table 2, the proposed method achieved the best average P-AUROC of 84.5%, which is improved over the second-place method Reg3D-AD by 17.7% and R3D-AD by 18.8% (author report: P-AUROC is 65.7%). Notably, for the "cap0" and "jar" categories, the P-AUROC values attained by our method are 96.8% and 96.7%, respectively, whereas Reg3D-AD only achieved 73.0% and 76.5%. This significant improvement could potentially be attributed to the ability of the proposed method to adapt samples with variations using a unified feature representation, thereby enhancing the discriminative power of the extracted features, particularly in sparse point clouds. Furthermore, the utilization of a large number of points for each group ($K = 512$) may have contributed to the superior anomaly detection performance in such sparse point cloud scenario.

2) Comparison on Real3D-AD. Table 3 shows the pixel-level detection performance of all methods on Real3D-AD. Our method achieved 78.3% P-AUROC on average. Among all selected methods, ours performs the best in terms of P-AUROC, surpassing the second-place method by 1.6%. Particularly, for the "Candybar" and "Diamond" categories, the P-AUROC values of our method are 91.3% and 97.5%, respectively. The reason for this may be attributed to different point clouds sharing similar structures may have large variations in orientation and position, leading to large differences in the extracted features, while our method exhibits great robustness in capturing such variations compared to other methods. Moreover, for large-scale point clouds like Real3D-AD, our method obtained superior representation and performance with fewer groups ($G = 512$) compared to Reg3D-AD ($G = 16384$) and IMRNet (downsampled to 8192 points). This suggests that our CTF-Net could extract better features for 3D anomaly detection.

4.3. Ablation Studies and Parameter Analysis

To evaluate the importance of key components and the effect of parameters on our method, we conducted ablation studies and parameter sensitivity analysis on Anomaly-ShapeNet, and the results are shown in Table 4.

Method	P-AUROC	O-AUROC	P-AUPRO	O-AUPRO	FPS	Memory
$\mathbf{Ours}_{w/o\ \mathbf{PCM}}$	0.5858	0.5028	0.0167	0.5196	0.31	25.85%
$\mathbf{Ours}_{w/o\ \mathbf{S3DA}}$	0.7884	0.6221	0.1732	0.6982	0.31	25.85%
$\mathbf{Ours}_{replace\ \mathbf{RANSAC}}$	0.7674	0.6732	0.1660	0.7260	0.29	25.85%
$\mathbf{Ours}_{replace\ \mathbf{PCA}}$	0.6141	0.5148	0.0178	0.5374	0.14	25.85%
$\mathbf{Ours}_{w/o\ \mathbf{Trans}}$	0.7793	0.6720	0.1622	0.7184	0.41	19.47%
$\mathbf{Ours}_{replace\ \mathbf{CCB}}$	0.8014	0.6213	0.1874	0.6844	0.50	16.98%
$\mathbf{Ours}_{CCB=1}$	0.7263	0.6210	0.1364	0.6824	0.42	19.11%
$\mathbf{Ours}_{CCB=2}$	0.7588	0.6223	0.1410	0.6963	0.39	19.94%
$\mathbf{Ours}_{CCB=3}$	0.8001	0.6327	0.1684	0.7163	0.36	21.92%
$\mathbf{Ours}_{CCB=4}$	0.8253	0.6609	0.2029	0.7432	0.35	23.33%
$\mathbf{Ours}_{CCB=5}$	**0.8446**	**0.6893**	**0.2124**	**0.7480**	0.31	25.85%
$\mathbf{Ours}_{G=64,K=512}$	0.6694	0.5960	0.0553	0.6655	0.50	16.34%
$\mathbf{Ours}_{G=128,K=512}$	0.7456	0.6070	0.1021	0.6826	0.43	19.68%
$\mathbf{Ours}_{G=256,K=512}$	0.8117	0.6692	0.1639	0.7318	0.36	20.43%
$\mathbf{Ours}_{G=512,K=64}$	0.7919	0.6127	0.1193	0.6788	1.54	12.23%
$\mathbf{Ours}_{G=512,K=128}$	0.7782	0.6297	0.0973	0.6809	1.02	14.42%
$\mathbf{Ours}_{G=512,K=256}$	0.8215	0.6487	0.1532	0.7087	0.68	16.84%
$\mathbf{Ours}_{G=512,K=512}$	**0.8446**	**0.6893**	**0.2124**	**0.7480**	0.31	25.85%

Table 4: Results of ablation and parameter sensitivity experiments, where w/o PCM, w/o S3DA, and w/o Trans. mean the proposed method without PCM, S3DA, and transformation matrix, respectively, replace RANSAC, replace PCA, and replace CCB mean replacing our PCM with RANSAC or PCA and replacing CCB with 1D convolution, respectively; Parameter CCB denotes the number of blocks in the feature extractor CTF-Net, G and K represent the number of groups and the number of points within each group, respectively. Frames Per Second (FPS) and Memory (%) stand for the average inference speed and memory consumption of our CTF-Net with an RTX3090, respectively.

As shown in Table 4, our method without the PCM component obtained only 58.58% P-AUROC, 50.28% O-AUROC, 1.67% P-AUPRO, and 51.96% O-AUPRO. However, after introducing the PCM component, our method ($G = 512, K = 512$) achieved a performance improvement by 25.88%, 18.65%, 19.57%, and 22.84% in terms of different evaluation metrics, respectively. The replacement of the PCM with the RANSAC registration in Reg3D-AD and commonly used PCA registration results in a significant degradation in performance. For instance, the PCA registration caused a 23.05% degradation in P-AUROC, 17.45% in O-AUROC, 19.46% in P-AUPRO, and 21.05% in O-AUPRO, respectively. The reason for this may be attributed to the fact that the PCM maps point clouds to a rotationally invariant space before feature extraction. For other methods, slight changes of point clouds in direction cause variations in extracted features, while the PCM eliminates the effect of orientation on data, thereby resulting in rotation-invariant features. This significantly improves the robustness of our method for 3D anomaly detection.

The proposed feature extractor CTF-Net, containing several CCB blocks and transformation matrix, was trained with the S3DA strategy. After removing S3DA, the performance was reduced by 5.62%, 6.72%, 3.92%, and 4.98%, respectively, in four metrics. This may be due to the fact that S3DA provides diverse training data to facilitate CTF-Net to learn more discriminative features. When removing the transformation matrix, the performance was decreased by 6.53%, 1.73%, 5.02%, and 2.96%, respectively, in different metrics. Moreover, after replacing the CCB module with 1D convolution, the performance was reduced by 4.32%, 6.80%, 2.50%, and 6.36%, respectively. Finally, reducing the number of CCB modules from 5 to 1 resulted in performance decrease of 11.83%, 6.83%, 7.60% and 6.56%, respectively, in four metrics. The reason may be that more CCB modules benefit

(a) P-AUROC													
Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Average
M3DM(PointBERT)	0.512	0.574	0.624	0.734	0.582	0.642	0.557	**0.643**	0.532	0.648	0.443	0.657	0.596
M3DM(PointMAE)	0.532	0.517	**0.698**	**0.684**	0.598	**0.661**	0.587	0.634	0.572	0.674	0.534	0.562	0.604
PatchCore(PointMAE)	0.547	0.608	0.684	0.667	0.725	0.487	0.662	0.538	**0.692**	**0.744**	**0.628**	0.662	0.637
Reg3D-AD	**0.654**	**0.711**	0.652	0.637	**0.758**	0.533	**0.749**	0.513	0.738	0.743	0.632	0.722	0.670
RIF (CTF-Net) (Ours)	**0.703**	**0.784**	**0.873**	0.742	**0.956**	0.827	0.732	0.847	0.554	0.692	0.624	**0.887**	0.768

(b) O-AUROC													
Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Average
M3DM(PointBERT)	0.387	0.422	0.495	0.575	0.427	0.434	0.487	**0.557**	0.502	0.537	0.508	0.570	0.492
M3DM(PointMAE)	0.527	**0.508**	0.482	0.560	0.612	0.438	0.530	**0.624**	**0.512**	**0.688**	0.520	0.562	0.547
PatchCore(PointMAE)	**0.711**	0.501	0.547	**0.797**	0.757	**0.514**	0.578	0.411	0.504	0.528	**0.524**	0.560	0.578
Reg3D-AD	0.602	**0.638**	0.604	0.668	0.768	0.490	0.677	0.344	**0.610**	0.538	0.469	0.588	0.583
RIF (CTF-Net) (Ours)	**0.608**	0.500	**0.624**	0.566	**0.824**	0.550	0.524	0.557	0.501	0.492	**0.580**	0.666	0.583

Table 5: One-Shot performance of different methods on Real3D-AD across 12 categories under Gaussian noise with a standard deviation of 0.002, where best results are highlighted in **bold**.

the capture of multi-scale structural features.

Our method involves two adjustable parameters: the number of groups G and the number of points K within each group. When varying the value of G from 64 to 512, the performance of our method increased greatly. Its P-AUROC, O-AUROC, P-AUPRO, and O-AUPRO were improved by 17.52%, 9.33%, 15.71%, and 8.25%, respectively. The explanation for this is that, as the number of groups grows, the number of features used to characterise each sample also increases, allowing for a more detailed description of the point cloud structure. When varying the value of K from 64 to 512, the P-AUROC, O-AUROC, P-AUPRO, and O-AUPRO of our method improved by 5.27%, 7.66%, 9.31%, and 6.92%, respectively. This performance improvement can be attributed to the fact that, as the number of points in each group increases, the local information hidden within the points is further exploited, enabling a more comprehensive understanding of the regional structure and its surrounding context within the point cloud. While the setting of $G = 512$ and $K = 512$ achieved a balance between computational efficiency and detection precision.

Our CTF-Net has only 4.1M parameters, much less than other methods such as PointMLP (12.6M), and memory consumption is only 26% on an RTX3090. In addition, our method achieved an FPS of 0.33, outperforming other methods such as Reg3D-AD (FPS of 0.13). We report in the Supplementary Material on the use of other datasets for pre-training CTF-Net to prevent potential data leakage

(a) P-AUROC													
Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Average
M3DM(PointBERT)	0.501	0.511	0.594	0.701	0.574	0.622	0.501	0.597	0.501	0.587	0.401	0.524	0.551
M3DM(PointMAE)	0.524	0.527	0.641	0.617	0.544	0.641	0.524	0.599	0.562	0.670	0.530	0.559	0.578
PatchCore(PointMAE)	0.533	0.574	0.633	0.614	0.711	0.492	0.627	0.527	**0.677**	**0.717**	**0.602**	0.622	0.611
Reg3D-AD	**0.658**	0.691	0.668	0.640	0.718	0.528	0.697	0.527	**0.698**	0.699	0.597	0.723	0.654
RIF (CTF-Net) (Ours)	0.644	**0.746**	**0.804**	**0.718**	**0.847**	**0.762**	**0.708**	**0.714**	0.478	0.617	0.600	**0.789**	**0.702**

(b) O-AUROC													
Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Average
M3DM(PointBERT)	0.400	0.418	0.463	0.548	0.458	0.455	0.472	0.548	0.479	0.501	0.501	0.562	0.484
M3DM(PointMAE)	0.501	0.504	0.472	0.547	0.601	0.452	0.517	**0.576**	0.520	**0.648**	0.498	0.542	0.532
PatchCore(PointMAE)	**0.681**	0.508	0.532	**0.728**	0.648	0.524	0.568	0.428	0.502	0.542	0.496	0.543	0.558
Reg3D-AD	0.578	**0.640**	**0.587**	0.643	0.614	0.501	**0.657**	0.341	**0.594**	0.501	0.428	0.571	0.555
RIF (CTF-Net) (Ours)	0.617	0.517	**0.630**	0.527	**0.727**	**0.552**	0.530	0.560	0.508	0.504	**0.542**	0.618	**0.569**

Table 6: One-Shot performance of different methods on Real3D-AD across 12 categories under Gaussian noise with a standard deviation of 0.003, where best and second-place results are highlighted in **bold**.

4.4. Robustness Experimental Results to Noisy Data.

The complexity of environments and equipment instability may result in scanned point clouds containing noise in real-world scenarios. To evaluate the performance of our model with noise, we conducted comparative experiments on point clouds with different standard Gaussian noises under one-shot conditions. For the fairness comparison, all pre-training was performed on ModelNet40 [32]. The results for Gaussian noise with a standard deviation of 0.002 and 0.003 are shown in Tables 5 and 6, respectively. It is observed that the performance decreases only slightly as the standard deviation of the noise increases, and is more noise-resistant compared to other approaches, showing better anomaly detection performance(P-AUROC at standard deviations of 0.02 and 0.03 was 9.8% and 4.8% higher than the baseline model Reg3D-AD [1], respectively). This may be attributed to the ability of rotationally invariant features to capture features more robustly under noisy conditions. These empirical results demonstrate the robustness of our method in detecting anomalies from noisy point clouds.

4.5. Generalization of Our Framework RIF

To examine the RIF's generalization, we conducted extensibility experiments on Anomaly-ShapeNet. The feature extractor CTF-Net was replaced by other representative point cloud feature extractors, including PointNet [36], PointMLP [37], DGCNN [38], PointMAE [37], Stratified-Transformer [39], and PointTransformer-v3 [40], and some results are reported in Table 7.

Method	P-AUROC	O-AUROC	P-AUPRO	O-AUPRO
RIF$_{w/o\ PCM}$ + PointNet	0.5081	0.5037	0.0177	0.5716
RIF$_{PCM}$ + PointNet	**0.8348**	0.6770	0.1964	0.7336
RIF$_{w/o\ PCM}$ + PointMLP	0.7433	0.5751	0.0679	0.6418
RIF$_{PCM}$ + PointMLP	0.8196	**0.7533**	**0.2987**	0.7950
RIF$_{w/o\ PCM}$ + DGCNN	0.6054	0.5200	0.0230	0.5857
RIF$_{PCM}$ + DGCNN	0.7658	0.7170	0.1963	**0.8389**
RIF$_{w/o\ PCM}$ + PointMAE	0.6442	0.5897	0.0437	0.6148
RIF$_{PCM}$ + PointMAE	0.7242	0.6880	0.0544	0.7522
RIF$_{w/o\ PCM}$ + Stratified-Transformer	0.6142	0.6142	0.0208	0.6428
RIF$_{PCM}$ + Stratified-Transformer	0.6357	0.6685	0.0574	0.7125
RIF$_{w/o\ PCM}$ + PointTransformer-v3	0.6328	0.5927	0.0411	0.6284
RIF$_{PCM}$ + PointTransformer-v3	0.7842	0.7110	0.2384	0.7842

Table 7: Results of RIF with other feature extractors on Anomaly-ShapeNet in terms of different evaluation metrics.

As shown in Table 7, the integration of RIF with other feature extractors greatly enhances anomaly detection performance. On Anomaly-ShapeNet, RIF with PCM achieved notable improvements over baseline (w/o PCM): PointNet (P-AUROC: +32.67%, O-AUROC: +17.33%), PointMLP (P-AUROC: +7.63%, O-AUROC: +17.82%), DGCNN (P-AUROC: +16.04%, O-AUROC: +19.70%), PointMAE (P-AUROC: +8.00%, O-AUROC: +9.83%), Stratified-Transformer (P-AUROC: +2.15%, O-AUROC: +5.43%), and PointTransformer-v3 (P-AUROC: +15.14%, O-AUROC: +11.83%).

For Real3D-AD, as shown in Table 8, RIF with PCM obtained significant performance improvement, particularly with PointNet (P-AUROC: +24.23%, O-AUROC: +8.80%), DGCNN (P-AUROC: +16.17%, O-AUROC: +3.15%), Stratified-Transformer (P-AUROC: +10.23%, O-AUROC: +3.14%), and PointTransformer-v3 (P-AUROC: +6.12%, O-AUROC: +2.30). These results demonstrate the strong adaptability of RIF to convolution-based (PointNet, DGCNN), MLP-based (PointMLP), and transformer-based (PointMAE, Stratified-Transformer, PointTransformer-v3) architectures, making it highly suitable for industrial 3D anomaly detection scenarios where available data exhibits a high degree of diversity and variability.

13

Real3D-AD				
Method	P-AUROC	O-AUROC	P-AUPRO	O-AUPRO
RIF$_{\text{w/o PCM}}$ + PointNet	0.5387	0.5132	0.0171	0.5433
RIF$_{\text{PCM}}$ + PointNet	**0.7810**	0.6012	**0.1053**	0.6035
RIF$_{\text{w/o PCM}}$ + PointMLP	0.7267	0.5365	0.0483	0.5693
RIF$_{\text{PCM}}$ + PointMLP	0.7673	0.5194	0.0548	0.5458
RIF$_{\text{w/o PCM}}$ + DGCNN	0.5962	0.4932	0.0273	0.5200
RIF$_{\text{PCM}}$ + DGCNN	0.7579	0.5247	0.0552	0.5403
RIF$_{\text{w/o PCM}}$ + PointMAE	0.6214	0.5430	0.0355	0.5398
RIF$_{\text{PCM}}$ + PointMAE	0.6478	0.5882	0.0398	0.6112
RIF$_{\text{w/o PCM}}$ + Stratified-Transformer	0.6447	0.6128	0.0543	0.6427
RIF$_{\text{PCM}}$ + Stratified-Transformer	0.7470	**0.6443**	0.0830	**0.6574**
RIF$_{\text{w/o PCM}}$ + PointTransformer-v3	0.6148	0.5883	0.0361	0.6044
RIF$_{\text{PCM}}$ + PointTransformer-v3	0.6760	0.6113	0.0743	0.6434

Table 8: Results of RIF with other feature extractors in terms of different evaluation metrics.

4.6. Efficiency Comparison between Our Method and Existing Methods

Inference speed and memory usage are very important in industrial applications [41]. As shown in Figure 3, our method with the rotationally invariant feature mechanism requires only a low memory capacity to achieve the best results. Compared to existing feature embedding approaches, our approach outperforms in both memory consumption (Memory: 26%) and inference speed (FPS: 0.31). This may be attributed to the use of a lightweight feature extraction network and fewer groups to construct the memory bank, with a significant reduction in the search space.
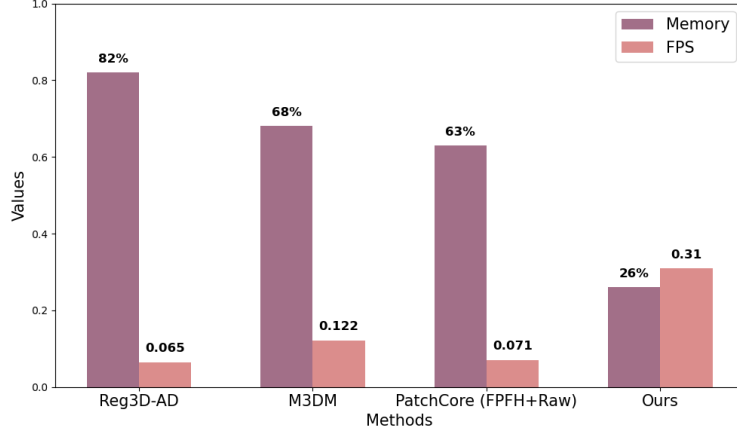

Figure 3: Comparison between our method with other methods in terms of inference speed and maximum memory during inference on Anomaly-ShapeNet, with experiments performed on an RTX3090.

5. Conclusion

This paper presented an innovative 3D anomaly detection (AD) framework RIF, aiming at addressing challenges from samples with orientation and position variations in real industrial production scenarios. To eliminate the negative effects of data variations on feature extraction, we proposed the Point Coordinate Mapping (PCM) technique to transform point clouds from any orientation into a consistent point cloud representation. Moreover, to learn from normal point cloud samples, we developed a Convolutional Transform Feature Network (CTF-Net) to derive highly discriminative features for the memory bank. Additionally, we introduced a Spatial 3D Data Augmentation

(S3DA) technique to aid the pre-training of our feature extraction networks. Experiments on the Anomaly-ShapeNet and Real3D-AD datasets show that our method achieved SOTA performance across different metrics. Our RIF framework is also highly adaptable to other feature extraction networks, providing a promising solution for industrial 3D AD.

6. Supplementary Material

6.1. The Proof of the Point Coordinate Mapping (PCM) for Rotational Invariance

6.1.1. Definitions

Definition A1: Given a point cloud $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^n \in \mathbb{R}^{n \times 3}$ with each point $\mathbf{p}_i = (x_i, y_i, z_i)^\top$, the centroid of $\mathcal{P}$ is defined as:

$$\mathbf{c} = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i. \tag{10}$$

Definition A2: Given a point cloud $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^n \in \mathbb{R}^{n \times 3}$, the relative coordinate vector of each point $\mathbf{p}_i (i=1, 2, \ldots, n)$ with respect to the centroid $\mathbf{c}$ is defined as:

$$\mathbf{r}_i = \mathbf{p}_i - \mathbf{c}. \tag{11}$$

Definition A3: Let $\mathbf{R} \in \mathrm{SO}(3)$ be a matrix with the size of 3*3. $\mathbf{R}$ is a unitized rotation matrix if and only if:
1) $\mathbf{R}\mathbf{R}^\top = \mathbf{I}_3$,
2) $Det(\mathbf{R}) = 1$,
where $\mathrm{SO}(3)$ denotes the 3D rotation group, and Det means the value of the determinant.

Definition A4: Given a rotation matrix $\mathbf{R}$, the transformed relative coordinate of each point $\mathbf{p}_i$ is defined as:

$$\mathbf{p}_i' = \mathbf{R}\mathbf{r}_i + \mathbf{c}. \tag{12}$$

Definition A5: Given a point cloud $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^n \in \mathbb{R}^{n \times 3}$, the three key vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ to define the relative coordinate system can be formalized as:

$$\begin{aligned}
\mathbf{u}_1 &= \mathbf{p}_{\mathrm{far1}} - \mathbf{c}, \\
\mathbf{u}_2 &= \mathbf{p}_{\mathrm{far2}} - \mathbf{c}, \\
\mathbf{u}_3 &= \mathbf{p}_{\mathrm{near}} - \mathbf{c}.
\end{aligned} \tag{13}$$

where $\mathbf{p}_{\mathrm{far1}}$ is the farthest point from the centroid, $\mathbf{p}_{\mathrm{far2}}$ is the second farthest point from the centroid and is linearly independent to $\mathbf{u}_1$, and $\mathbf{p}_{\mathrm{near}}$ is the nearest point to the centroid that makes the rank of the three selected vectors equal to 3, i.e., $Rank([\mathbf{u}_1; \mathbf{u}_2; \mathbf{u}_3]) = 3$.

Definition A6: Given three key vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$, the orthogonalized vectors after Schmidt orthogonalization can be defined as:

$$\begin{aligned}
\mathbf{e}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}, \\
\mathbf{e}_2 &= \frac{\mathbf{u}_2 - (\mathbf{u}_2 \cdot \mathbf{e}_1)\mathbf{e}_1}{\|\mathbf{u}_2 - (\mathbf{u}_2 \cdot \mathbf{e}_1)\mathbf{e}_1\|}, \\
\mathbf{e}_3 &= \frac{\mathbf{u}_3 - (\mathbf{u}_3 \cdot \mathbf{e}_1)\mathbf{e}_1 - (\mathbf{u}_3 \cdot \mathbf{e}_2)\mathbf{e}_2}{\|\mathbf{u}_3 - (\mathbf{u}_3 \cdot \mathbf{e}_1)\mathbf{e}_1 - (\mathbf{u}_3 \cdot \mathbf{e}_2)\mathbf{e}_2\|}.
\end{aligned} \tag{14}$$

Definition A7: Given a point cloud $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^n \in \mathbb{R}^{n \times 3}$, the **P**oint **C**oordinate **M**apping (**PCM**) with rotational invariance and the relative coordinate point after mapping are defined as:

$$\mathbf{M} : \mathbf{P} \to \mathbf{P}^*. \tag{15}$$

$$\mathbf{p}_i^* = (\mathbf{p}_i - \mathbf{c})[\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3]^{\mathrm{T}}. \tag{16}$$

where $\mathbf{p}_i^*$ represents the point with rotational invariance after being processed by **PCM**.

6.1.2. Propositions

Proposition A1: The coordinate representation of the centroid of the point cloud $\mathcal{P}$ is rotationally invariant.

Proof. Given a rotation matrix $\mathbf{R}$ satisfied **Definition A3**, the centroid $\mathbf{c}'$ of the rotated point cloud $\mathcal{P}' = \{\mathbf{p}_i'\}_{i=1}^n$ satisfied **Definition A4** has the following property:

$$
\begin{aligned}
\mathbf{c}' &= \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i', \\
&= \frac{1}{n} \sum_{i=1}^n \left(\mathbf{R}(\mathbf{p}_i - \mathbf{c}) + \mathbf{c} \right), \\
&= \mathbf{R} \left(\frac{1}{n} \sum_{i=1}^n (\mathbf{p}_i - \mathbf{c}) \right) + \mathbf{c}, \\
&= \mathbf{c}.
\end{aligned}
\tag{17}
$$

Thus, the centroid of the point cloud is rotationally invariant. The proof of **Proposition 1** is complete. $\square$

Proposition A2: the orthogonalized vectors $\mathbf{e}_1', \mathbf{e}_2', \mathbf{e}_3'$ are simply the original orthogonalized vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ transformed by the rotation matrix $\mathbf{R}$.

Proof. Given the relative position vectors $\mathbf{r}_i (i = 1, 2, \ldots, n)$ satisfied **Definition A2** and $\mathbf{R} \in \mathrm{SO}(3)$ satisfied **Definition A3**, it has:

$$
\begin{aligned}
\mathbf{r}_i' &= \mathbf{p}_i' - \mathbf{c}', \\
&= \mathbf{R}(\mathbf{p}_i - \mathbf{c}), \\
&= \mathbf{R}\mathbf{r}_i.
\end{aligned}
\tag{18}
$$

Given three key vectors $\mathbf{u}_1, \mathbf{u}_2,$ and $\mathbf{u}_3$ satisfied **Definition A5** and $\mathbf{R} \in \mathrm{SO}(3)$ satisfied **Definition A3**, these vectors after rotation can be represented as:

$$
\begin{aligned}
\mathbf{u}_1' &= \mathbf{R}\mathbf{u}_1, \\
\mathbf{u}_2' &= \mathbf{R}\mathbf{u}_2, \\
\mathbf{u}_3' &= \mathbf{R}\mathbf{u}_3.
\end{aligned}
\tag{19}
$$

After Schmidt Orthogonalization, these vectors can be represented as:

$$
\begin{aligned}
\mathbf{e}_1' &= \frac{\mathbf{u}_1'}{\|\mathbf{u}_1'\|}, \\
&= \frac{\mathbf{R}\mathbf{u}_1}{\|\mathbf{R}\mathbf{u}_1\|}, \\
&= \mathbf{R}\left(\frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}\right) = \mathbf{R}\mathbf{e}_1.
\end{aligned}
\tag{20}
$$

$$\begin{aligned}
\mathbf{e}_2' &= \frac{\mathbf{u}_2' - (\mathbf{u}_2' \cdot \mathbf{e}_1')\mathbf{e}_1'}{\|\mathbf{u}_2' - (\mathbf{u}_2' \cdot \mathbf{e}_1')\mathbf{e}_1'\|}, \\
&= \frac{\mathbf{R}\mathbf{u}_2 - (\mathbf{u}_2 \cdot \mathbf{e}_1)\mathbf{R}\mathbf{e}_1}{\|\mathbf{R}(\mathbf{u}_2 - (\mathbf{u}_2 \cdot \mathbf{e}_1)\mathbf{e}_1)\|}, \\
&= \mathbf{R}\left(\frac{\mathbf{u}_2 - (\mathbf{u}_2 \cdot \mathbf{e}_1)\mathbf{e}_1}{\|\mathbf{u}_2 - (\mathbf{u}_2 \cdot \mathbf{e}_1)\mathbf{e}_1\|}\right) = \mathbf{R}\mathbf{e}_2.
\end{aligned} \tag{21}$$

Similarly,

$$\mathbf{e}_3' = \mathbf{R}\mathbf{e}_3. \tag{22}$$

Thus, the orthogonalized vectors $\mathbf{e}_1', \mathbf{e}_2', \mathbf{e}_3'$ are simply the original orthogonalized vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ transformed by the rotation matrix $\mathbf{R}$. The proof of **Proposition 2** is complete.

$\square$

Proposition A3: Give a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ and point cloud $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^n \in \mathbb{R}^{n \times 3}$, the pre-rotation and post-rotation representations of each point $\mathbf{p}_i = (x_i, y_i, z_i)^T (i=1, 2, \ldots, n)$ being processed by **PCM** satisfies:

$$(x_i, y_i, z_i)_{\text{pre-rotation}} \equiv (x_i, y_i, z_i)_{\text{post-rotation}}. \tag{23}$$

Proof. Given the original point cloud processed by **PCM** satisfied **Definition A7**, *i.e.*, $\mathbf{p}_i^* = (\mathbf{p}_i - \mathbf{c}) \cdot [\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3]^\top$. For the rotated point cloud $\mathbf{p}_i' = \mathbf{R}(\mathbf{p}_i - \mathbf{c}) + \mathbf{c}$ satisfied **Definition A4**, the computation of its PCM coordinates can be formed as follows:

1) Obtain the relative coordinate centroid (**Proposition A1**): $\mathbf{c}' = \mathbf{c}$.

2) Generate the orthogonal basis vectors (**Proposition A2**): The rotated basis vectors satisfy $[\mathbf{e}_1'; \mathbf{e}_2'; \mathbf{e}_3'] = \mathbf{R}[\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3]$.

3) Perform **PCM** mapping for rotated points:

$$\begin{aligned}
\mathbf{p}_i^{*\prime} &= \mathbf{R}(\mathbf{p}_i - \mathbf{c}) \cdot (\mathbf{R}[\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3])^\top, \\
&= \mathbf{R}\mathbf{R}^T(\mathbf{p}_i - \mathbf{c}) \cdot [\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3]^\top, \\
&= (\mathbf{p}_i - \mathbf{c}) \cdot [\mathbf{e}_1; \mathbf{e}_2; \mathbf{e}_3]^\top, \\
&= \mathbf{p}_i^*
\end{aligned} \tag{24}$$

Thus, **PCM** is rotationally invariant. The proof of **Proposition 3** is complete. $\square$

6.1.3. Conclusion

The PCM-processed coordinates $\mathbf{p}_i^*$ are invariant under any rotation $\mathbf{R} \in \mathrm{SO}(3)$, as the transformed coordinates $\mathbf{p}_i^{*\prime}$ equal $\mathbf{p}_i^*$. Thus, **PCM** has the property of rotational invariance.

7. Acknowledge

References

[1] J. Liu, G. Xie, R. Chen, X. Li, J. Wang, Y. Liu, C. Wang, and F. Zheng, "Real3d-ad: A dataset of point cloud anomaly detection," in *Advances in Neural Information Processing Systems* (A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds.), vol. 36, pp. 30402–30415, 2023.

[2] M. Kruse, M. Rudolph, D. Woiwode, and B. Rosenhahn, "Splatpose & detect: Pose-agnostic 3d anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 3950–3960, June 2024.

[3] H. Zhu, G. Xie, C. Hou, T. Dai, C. Gao, J. Wang, and L. Shen, "Towards high-resolution 3d anomaly detection via group-level feature contrastive learning," in *Proceedings of the 32nd ACM International Conference on Multimedia*, p. 4680–4689, 2024.

[4] W. Li, X. Xu, Y. Gu, B. Zheng, S. Gao, and Y. Wu, "Towards scalable 3d anomaly detection and localization: A benchmark via 3d anomaly synthesis and a self-supervised learning network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22207–22216, June 2024.

[5] L. Xu, D. Han, G. Li, M. Zhou, J. Wan, and M. Li, "Multimodal feature cooperative refinement for few-shot anomaly detection," *Advanced Engineering Informatics*, vol. 68, p. 103792, 2025.

[6] Y. Chen, D. He, S. He, Z. Jin, J. Miao, S. Shan, and Y. Chen, "Welding defect detection based on phased array images and two-stage segmentation strategy," *Advanced Engineering Informatics*, vol. 62, p. 102879, 2024.

[7] E. Horwitz and Y. Hoshen, "Back to the feature: Classical 3d features are (almost) all you need for 3d anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2968–2977, June 2023.

[8] Z. Zhou, L. Wang, N. Fang, Z. Wang, L. Qiu, and S. Zhang, "R3d-ad: Reconstruction via diffusion for 3d anomaly detection," in *Computer Vision – ECCV 2024* (A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, eds.), pp. 91–107, 2025.

[9] Y. Wang, J. Peng, J. Zhang, R. Yi, Y. Wang, and C. Wang, "Multimodal industrial anomaly detection via hybrid fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8032–8041, June 2023.

[10] P. Bergmann, X. Jin, D. Sattlegger, and C. 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*, 2022.

[11] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), pp. 604–621, 2022.

[12] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] R. Lu, Y. Wu, L. Tian, D. Wang, B. Chen, X. Liu, and R. Hu, "Hierarchical vector quantized transformer for multi-class unsupervised anomaly detection," in *Advances in Neural Information Processing Systems* (A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds.), vol. 36, pp. 8487–8500, Curran Associates, Inc., 2023.

[14] Y. Zhao, "Omnial: A unified cnn framework for unsupervised anomaly localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3924–3933, June 2023.

[15] M. Asad, W. Azeem, A. A. Malik, H. Jiang, A. Ali, J. Yang, and W. Liu, "3d-mmfn: Multi-level multimodal fusion network for 3d industrial image anomaly detection," *Advanced Engineering Informatics*, vol. 65, p. 103284, 2025.

[16] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, mar 2021.

[17] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[18] Y. Liang, J. Zhang, S. Zhao, R. Wu, Y. Liu, and S. Pan, "Omni-frequency channel-selection representations for unsupervised anomaly detection," *IEEE Transactions on Image Processing*, vol. 32, pp. 4327–4340, 2023.

[19] H. He, J. Zhang, H. Chen, X. Chen, Z. Li, X. Chen, Y. Wang, C. Wang, and L. Xie, "A diffusion-based framework for multi-class anomaly detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 8472–8480, Mar. 2024.

[20] V. Zavrtanik, M. Kristan, and D. Skočaj, "Reconstruction by inpainting for visual anomaly detection," *Pattern Recognition*, vol. 112, p. 107706, 2021.

[21] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14318–14328, June 2022.

[22] X. Jiang, J. Liu, J. Wang, Q. Nie, K. WU, Y. Liu, C. Wang, and F. Zheng, "Softpatch: Unsupervised anomaly detection with noisy data," in *Advances in Neural Information Processing Systems* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 15433–15445, 2022.

[23] V. Zavrtanik, M. Kristan, and D. Skočaj, "DrÆm – a discriminatively trained reconstruction embedding for surface anomaly detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8310–8319, 2021.

[24] J. C. Lee, T. Kim, E. Park, S. S. Woo, and J. H. Ko, "Continuous memory representation for anomaly detection," in *Computer Vision – ECCV 2024* (A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, eds.), pp. 438–454, 2025.

[25] C. Ding, G. Pang, and C. Shen, "Catching both gray and black swans: Open-set supervised anomaly detection," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7378–7388, 2022.

[26] H. Zhang, Z. Wu, Z. Wang, Z. Chen, and Y.-G. Jiang, "Prototypical residual networks for anomaly detection and localization," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16281–16291, 2023.

[27] J. Zhu, C. Ding, Y. Tian, and G. Pang, "Anomaly heterogeneity learning for open-set supervised anomaly detection," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17616–17626, 2024.

[28] G. Chen, C. G. Koh, S. Li, Z. Jiang, Y. Liang, Z. Xu, and D. Wu, "Intelligent integrity detection and damage localization of pile from low-strain test using deep learning with accelerated training via variable-order gradient descent," *Advanced Engineering Informatics*, vol. 68, p. 103690, 2025.

[29] Y. Cao, X. Xu, and W. Shen, "Complementary pseudo multimodal feature for point cloud anomaly detection," *Pattern Recognition*, vol. 156, p. 110761, 2024.

[30] A. Bhunia, C. Li, and H. Bilen, "Looking 3d: Anomaly detection with 2d-3d alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17263–17272, June 2024.

[31] H. Yue, Q. Wang, H. Huang, X. Xia, H. Fang, and J. C. Cheng, "Enhancing semantic segmentation of mep scenes with deep learning and bim-generated synthetic point clouds," *Advanced Engineering Informatics*, vol. 68, p. 103723, 2025.

[32] J. Sun, Q. Zhang, B. Kailkhura, Z. Yu, C. Xiao, and Z. M. Mao, "Modelnet40-c: a robustness benchmark for 3d point cloud recognition under corruption," in *ICLR 2022 Workshop on Socially Responsible Machine Learning*, vol. 7, 2022.

[33] J. Cheng, C. Gao, J. Zhou, J. Wen, T. Dai, and J. Wang, "Mc3d-ad: A unified geometry-aware reconstruction model for multi-category 3d anomaly detection," 2025.

[34] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3d point cloud registration," 2021.

[35] C. He, Z. Zhao, X. Zhang, H. Yu, and R. Wang, "Rotinv-pct: Rotation-invariant point cloud transformer via feature separation and aggregation," *Neural Networks*, vol. 185, p. 107223, 2025.

[36] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017.

[37] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," 2022.

[38] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, 2019.

[39] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia, "Stratified transformer for 3d point cloud segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8500–8509, 2022.

[40] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point transformer v3: Simpler faster stronger," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4840–4851, 2024.

[41] Y.-M. Chu, C. Liu, T.-I. Hsieh, H.-T. Chen, and T.-L. Liu, "Shape-guided dual-memory learning for 3D anomaly detection," in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, pp. 6185–6194, 23–29 Jul 2023.