Taming Anomalies with Down-Up Sampling Networks: Group Center Preserving Reconstruction for 3D Anomaly Detection

Hanzhe Liang
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen Audencia Financial Technology Institute
Shenzhen, China
2023362051@email.szu.edu.cn

Jie Zhang
Faculty of Applied Sciences, Macao Polytechnic University
Macao, China
jpeter.zhang@mpu.edu.mo

Tao Dai
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
daitao@szu.edu.cn

Linlin Shen
School of Artificial Intelligence, Shenzhen University
Guangdong Provincial Key Laboratory of Intelligent Information Processing
Shenzhen, China
llshen@szu.edu.cn

Jinbao Wang
School of Artificial Intelligence, Shenzhen University
Guangdong Provincial Key Laboratory of Intelligent Information Processing
Shenzhen, China
wangjb@szu.edu.cn

Can Gao†
College of Computer Science and Software Engineering, Shenzhen University
Guangdong Provincial Key Laboratory of Intelligent Information Processing
Shenzhen, China
davidgao@szu.edu.cn

Abstract

Reconstruction-based methods have demonstrated very promising results for 3D anomaly detection. However, these methods face great challenges in handling high-precision point clouds due to the large scale and complex structure. In this study, a Down-Up Sampling Networks (DUS-Net) is proposed to reconstruct high-precision point clouds for 3D anomaly detection by preserving the group center geometric structure. The DUS-Net first introduces a Noise Generation module to generate noisy patches, which facilitates the diversity of training data and strengthens the feature representation for reconstruction. Then, a Down-sampling Network (Down-Net) is developed to learn an anomaly-free center point cloud from patches with noise injection. Subsequently, an Up-sampling Network (Up-Net) is designed to reconstruct high-precision point clouds by fusing multi-scale up-sampling features. Our method leverages group centers for construction, enabling the preservation of geometric structure and providing a more precise point cloud. Extensive experiments demonstrate the effectiveness of our proposed method, achieving state-of-the-art (SOTA) performance, with an Object-level AUROC of 79.9% and 79.5% and a Point-level AUROC of 71.2% and 84.7% on the Real3D-AD and Anomaly-ShapeNet datasets, respectively.

CCS Concepts

• **Computing methodologies** → **Visual inspection**.

1 Introduction

3D anomaly detection (AD) has attracted the spotlight of recent research due to the potential for high-precision industrial product inspection. The task requires the identification of points or regions that deviate from the normal distribution in a given 3D point cloud data. In real-world industrial applications, due to collecting normal samples is easy and labeling anomalous samples is time-consuming and laborious, unsupervised methods that learn only from normal

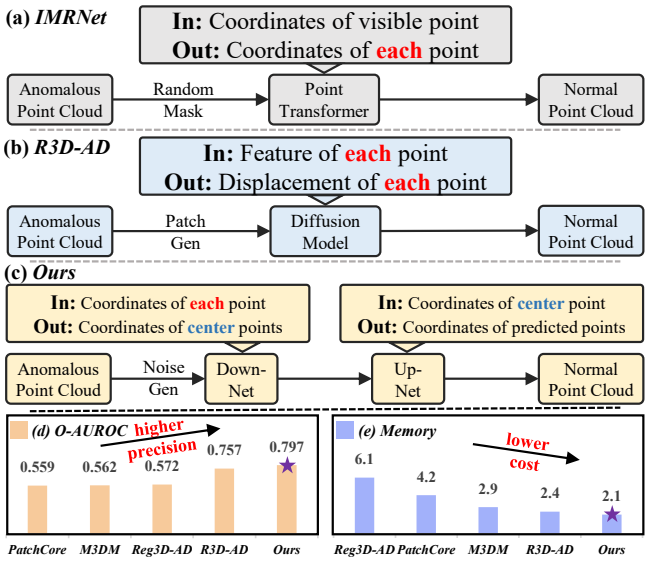

Figure 1: Comparison between previous methods and ours. The previous methods IMRNet [1] (a) and R3DAD [2] (b) predict the exact coordinates of each point, and our proposed (c) reconstruct a high-precision point cloud from precisely predicted group center points. This leads to higher performance (d) and faster inference speed (e).

This work was done by Hanzhe Liang at Shenzhen University, please email Hanzhe for any questions.

samples are widely used. Recent methods are roughly classified into feature embedding and feature reconstruction.

Feature embedding-based methods use pre-trained models or mathematical descriptors to extract features from normal samples and create a memory bank to match the features of the test samples, where points with large match errors are considered as anomalies. Some methods like BTF [3], AST [4], M3DM [5], and 3D-ADNAS [6] achieved excellent performance on MvTec3D-AD datasets [7]. Moreover, by exploiting robust feature extractors, some methods like Reg3D-AD [8], Group3AD [9], and ISMP [10] also attained impressive performance on more challenging datasets such as Anomaly-ShapeNet [1] and Real3D-AD [8]. Nevertheless, these feature embedding methods may have limitations in accurately extracting and representing features for each point.

The feature reconstruction methods encode the point cloud data into a latent space and decode into reconstruction feature space or coordinate space, with high reconstruction errors indicating the presence of anomalies. Some methods based on features, such as Shape-Guided [11] and CFM [12], exhibited their effectiveness in distinguishing between normality and abnormality. On the other hand, some point-based accurate reconstruction methods, like IMRNet [1] and R3D-AD [2], further improved the ability to detect point-level anomalies by predicting the coordinates or displacements of points. However, when handling high-precision point clouds, directly predicting point-level coordinates or displacements may face great challenges because of the large scale of points and complex structure. Grouping high-precision point clouds is a common technique used in recent methods, and the grouping centers maintain the main geometric structure of the point cloud while having a very small scale. Intuitively, group center preserving reconstruction is one of the potential alternatives, which maintains the overall geometric structure and also allows for up-sampling to restore high-precision point clouds. The differences between our solution and previous methods are shown in Figure 1.

Motivated by the above facts, we propose a Down-Up Sampling Networks (DUS-Net). Specifically, we first introduce a Noise Generation to generate noisy patches. Instead of directly predicting point-level information, we present a Down-sampling Network (Down-Net) to predict anomaly-free group centers from noisy patches to maintain overall geometric structures. Then, to meet the demand for high-precision anomaly detection, we design an Upsampling Network (Up-Net) to reconstruct high-precision normal point clouds by fusing multi-scale up-sampling features. The main contributions of this study are summarized as follows:

(1) To perform anomaly detection on high-precision point clouds, we propose a Down-Up Sampling Networks (DUS-Net), which allows for reconstructing high-precision and anomaly-free point clouds by capitalizing on group center-level points.
(2) To keep the overall geometric structure of the point clouds, we introduce a Down-Net to predict the group centers from noisy patches generated by a noise generation module, which provides the ability of denoising and also preserves the accurate topology structure points for reconstruction.
(3) To reconstruct high-precision point clouds, we present an Up-Sampling Net (Up-Net) to form multi-scale representations from down-sampled group center point clouds for reconstruction, which fuses multi-scale feature information and enable the generation of high-precision anomaly-free point clouds.
(4) Comparative experiments show that our method outperforms previous approaches and achieves SOTA performance, with an Object-level AUROC of 79.9% and 79.5% and a Point-level AUROC of 71.2% and 84.7% on Real3D-AD and Anomaly-ShapeNet, respectively.

2 Related Work

2.1 2D Anomaly Detection

2D anomaly detection is one of the crucial vision tasks, with the objective of detecting and localizing anomalies from images or patches [13, 14]. Current 2D anomaly detection methods predominantly follow two distinct paradigms: generative and discriminative methods [15]. The former leverage neural models such as Autoencoder [16, 17], Generative Adversarial Networks [18], and Diffusion [19] to capture representations or distributions of normal samples and identify anomalies by comparing with a memory bank [20, 21] or by reconstructing images with errors [22]. Methods like SPADE [23], DRAEM [24], and PatchCore [20] have achieved very impressive detection results on publicly available datasets. The latter employs supervised learning mechanisms and focuses on training classifiers to distinguish between defective and normal images with the aid of labeled images. Recent approaches such as DRA [25], BGAD [26], and AHL [27] validate the advantages of supervised information in enhancing detection precision. Notably, directly migrating 2D anomaly detection methods to 3D point clouds may encounter significant challenges due to inherent dimensional and structural disparities between data modalities.

2.2 3D Anomaly Detection

The 3D Anomaly Detection aims to detect and locate anomalous points or areas within 3D point cloud data [1, 10, 28]. Existing methods for 3D anomaly detection can be classified into two main categories: feature embedding and reconstruction methods.

Feature embedding methods employ pre-trained models or mathematical descriptors to extract representative features of normal samples for forming a memory bank, and anomalies are detected by comparing the features of the test sample with those in the memory bank. Reg3D-AD [8] used PointMAE [29] to extract registered features by RANSAC [30] and created a dual memory bank with patch features and point coordinates to compute point anomaly scores during inference. Group3AD [9] introduced a contrastive learning approach that maps different groups into distinct clusters to extracts group-level features for better anomaly detection. Looking3D [31], CPMF [32], and ISMP [10] extracted more discriminating features from generated 2D modalities as additional information to improve detection. Moreover, some multi-modal methods, such as BTF [3], M3DM [5], Shape-Guided [11], CFM [12], and 3D-ADNAS [6], achieved better anomaly detection by fusing features from point clouds and RGB images. Recently, the anomaly detection approaches based on the Large Language Model (LLM) also showed good detection performance on the zero-shot task [33–37].

Reconstruction methods encode the point cloud into a latent space and decode it back to its original form, where points with high reconstruction errors are regarded as anomalies. IMRNet [1] randomly masked and predicted the corresponding invisible coordinates to detect anomalies. R3D-AD [2] employed the diffusion model to compute the displacement of points for anomaly detection. Moreover, Splatpose [38] and Splatpose++ [39] used the Gaussian splatting to achieve multi-view reconstruction and anomaly detection. Although these methods achieve promising results, they may face great challenges in high-precision point clouds, which exhibit the characteristics of large scale and high complexity.

3 Approach

Reconstruction-based methods [1, 2] have shown their effectiveness in 3D anomaly detection. Nevertheless, when confronted with high-precision point clouds, these methods may have limitations in reconstruction quality and computational efficiency. To address these challenges, we propose the Down-Up Sampling Networks (DUS-Net) to reconstruct high-precision point clouds by preserving the group centers, and the overall framework is illustrated in Figure 2. It consists of three components: a Noise generation module, a center-preserved down-sampling module, and a multi-scale up-sampling module. Each component is described in detail in the following sections.

3.1 Patch Augmentation with Noise Generation

For 3D anomaly detection, the available training data is often very limited. To enrich the training data, some methods rely on generating pseudo-anomalies to improve their representation and detection capability. Nevertheless, the quality of the generated anomalies is difficult to guarantee, thereby affecting the performance. To address this problem, we propose a noise generation module (Noise-Gen) to diversify the data and promote the model to learn the ability of reconstructing normal points from noisy ones.

Given a point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$, we partition it into G patches $\{P_i\}_{i=1}^{G}$ via farthest point sampling (FPS) and K-nearest neighbors (KNN). Each patch $P_i \in \mathbb{R}^{(K+1)\times 3}$ is composed by $P_i = \{c_i \cup \mathcal{N}_K(c_i)\}$, where c_i denotes the center point selected by the FPS, and $\mathcal{N}_K(c_i)$ means the K nearest neighbors of c_i. To diversify the point cloud, Gaussian noises are injected into each group with two controllable noise parameters α and β. The patch after noise injection can be formulated as

$$\begin{aligned}\tilde{P}_i &= GSN(P_i, \alpha, \beta),\\ &= \{GSN(c_i, \alpha) \cup GSN(\mathcal{N}_K(c_i), \beta)\},\\ &= \{\tilde{c}_i \cup \tilde{\mathcal{N}}_K(c_i)\},\end{aligned} \tag{1}$$

where $GSN(\cdot,\cdot)$ denotes the operator of injecting at a certain level of Gaussian noise on a point or a set of points, and $\tilde{c}_i$ and $\tilde{\mathcal{N}}_K(c_i)$ represent the center and its nearest neighbors after injection of Gaussian noise.

After the noise injection process, points within each group are subjected to different levels of Gaussian noise. Despite causing subtle distortions to the structure, this process enriches the point cloud and facilitates a robust representation for downstream tasks.

3.2 Center-preserved Down-sampling with Down-Net

Point cloud data exhibits the characteristics of large scale and complexity. Directly reconstructing or predicting each point may face great challenges. To precisely reconstruct the point cloud, we propose a Down-Net to extract robust representations from noisy patches while preserving group centers.

Specifically, the noisy patches after injecting Gaussian noise are input into a Point-Net [40] to extract key features. While their centers are encoded as position embeddings by using a multilayer perceptron (MLP) to avoid leakage of the center information. These processes can be formalized as

$$\begin{aligned}E_c &= MLP(\tilde{c}_i),\ E_c \in \mathbb{R}^{G\times C_1};\\ E_p &= PointNet(\tilde{\mathcal{N}}_K(c_i)),\ E_p \in \mathbb{R}^{G\times C_2}.\end{aligned} \tag{2}$$

where MLP and $PointNet$ mean the feature extractor MLP and Point-Net, respectively, C_1 and C_2 denote the dimensions of the position embeddings and patch feature embeddings, respectively.

These patch embeddings, along with the position embeddings, are then fed into a transformer-architecture decoder to learn group-level features, where the attention mechanism captures the relationship of different patches to assist the decoding of each patch. The decoding process can be formalized as

$$E_f = Decoder(Concat(E_c, E_p)),\ E_f \in \mathbb{R}^{G\times C_3}, \tag{3}$$

where $Concat$ denotes the concatenation operator of different feature embeddings, $Decoder$ represents the decoder process, and C_3 stands for the dimension of the decoded patch features.

Finally, an MLP is used to predict centers, followed by a reshape operation to restore the predicted centers into the form of a 3D point cloud. The center prediction process can be represented as

$$\tilde{\mathcal{P}}_c = Reshape(MLP(E_f)), \tilde{\mathcal{P}}_c \in \mathbb{R}^{G\times 3}. \tag{4}$$

To precisely predict all centers under noise conditions, the Mean Squared Error (MSE) loss is used to minimize the errors between the original and predicted centers and to promote the learning of a robust feature representation for the group center geometric structure of the point cloud [41], which can be defined as

$$\mathcal{L}_{MSE}(\mathcal{P}_c, \tilde{\mathcal{P}}_c) = \frac{1}{N}\sum_{p_i\in\mathcal{P}_c}\sum_{p_j\in\tilde{\mathcal{P}}_c} \parallel p_i - p_j \parallel_2^2, \tag{5}$$

where $\mathcal{P}_c$ and $\tilde{\mathcal{P}}_c$ denote the original and predicted group center set, respectively, and $\|\cdot\|$ represents the 2-norm between two points.

To keep directional consistency on the predicted centers, the cosine similarity loss is further introduced, which is defined as

$$\mathcal{L}_{COS}(\mathcal{P}_c, \tilde{\mathcal{P}}_c) = \frac{1}{N}\sum_{p_i\in\mathcal{P}_c}\sum_{p_j\in\tilde{\mathcal{P}}_c}\left(1 - \frac{p_i \cdot p_j}{\parallel p_i \parallel_2 \parallel p_j \parallel_2}\right), \tag{6}$$

Moreover, the Chamfer distance loss [42] is employed to ensure macro-scale structural coherence, which is defined as

$$\begin{aligned}\mathcal{L}_{CD}(\mathcal{P}_c, \tilde{\mathcal{P}}_c) = &\frac{1}{\mathcal{P}_c}\sum_{p_i\in\mathcal{P}_c}\min_{p_j\in\tilde{\mathcal{P}}_c}\left\|p_i - p_j\right\|_2^2\\ &+\frac{1}{\tilde{\mathcal{P}}_c}\sum_{p_j\in\tilde{\mathcal{P}}_c}\min_{p_i\in\mathcal{P}_c}\left\|p_j - p_i\right\|_2^2.\end{aligned} \tag{7}$$

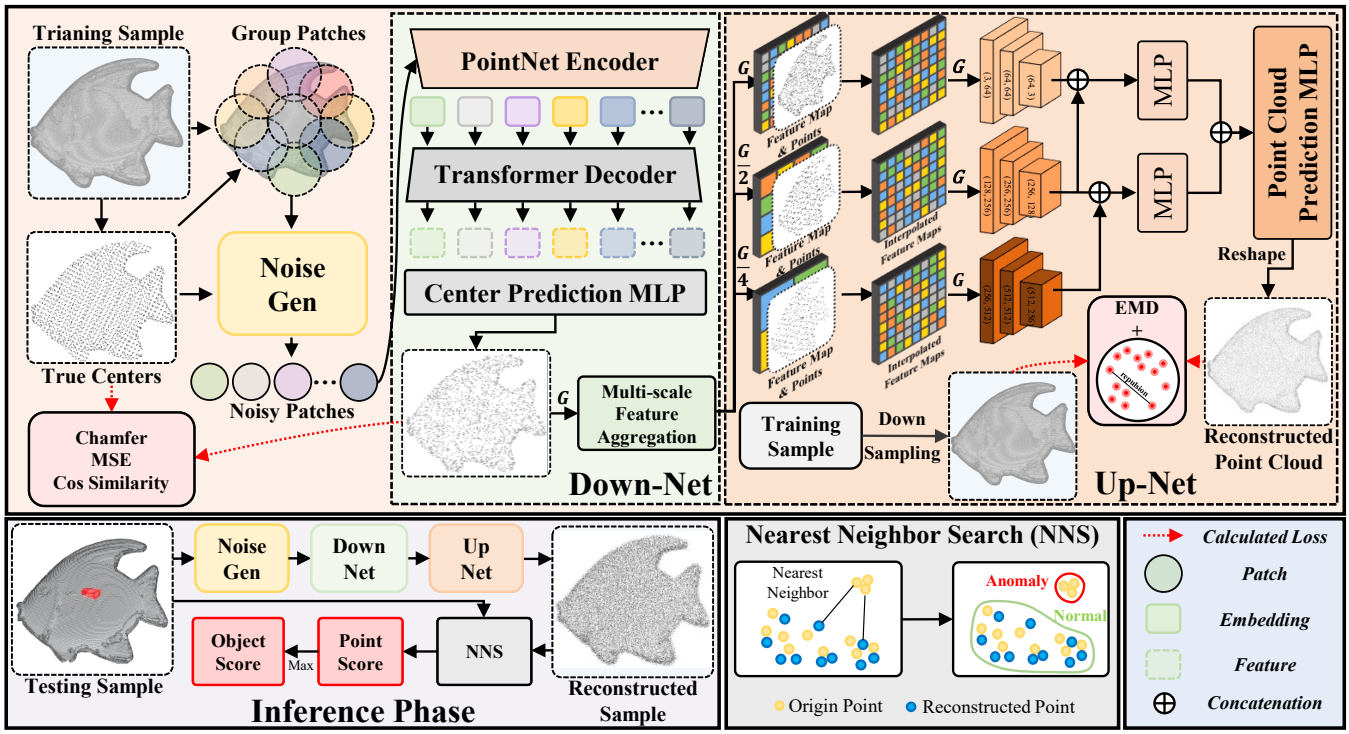

Figure 2: The pipeline of the proposed DUS-Net. The input point cloud is grouped by farthest point sampling (FPS) and K-nearest neighbor search, followed by noise injection using the Noise-Gen module. Then, the Down-Net is trained to predict anomaly-free center points from noisy patches. Finally, the Up-Net is used to reconstruct the predicted center point cloud into a high-precision anomaly-free point cloud by fusing multi-scale features, and points with large reconstruction errors are considered as anomalies.

The Chamfer distance adopts a bidirectional matching manner, with the first term ensuring the existence of corresponding predicted centers for each original center and the second term keeping the proximity of each predicted center to the original centers. Thus, the overall loss to optimize the Down-Net can be expressed as

$$\mathcal{L}_{Down} = \mathcal{L}_{MSE} + \mathcal{L}_{COS} + \mathcal{L}_{CD}. \tag{8}$$

3.3 Multiscale Reconstruction with Up-Net

Down-Net aims to predict centers for noisy patches to keep the main geometric structure of a point cloud. While these center-level points are not sufficient for high-precision anomaly detection. To address this problem, we propose an Up-Net to reconstruct high-precision point clouds by fusing multi-scale feature representations.

Specifically, the anomaly-free center-level point cloud $\tilde{\mathcal{P}}_c$ predicted by Down-Net is further processed by the feature aggregation technique in PointNet++ [43] to extract multi-scale features from the down-sampled point clouds with the scale of 1, 1/2, and 1/4, respectively. Formally, the features extracted in the l-th scale can be defined as [43]

$$F_i^l = \underset{p_j^{l-1} \in \mathcal{N}^{l-1}(p_i^l)}{MaxPool} \left(MLP\left([p_j^{l-1} - p_i^l; F_j^{l-1}] \right) \right), 1 \le l \le 2, \tag{9}$$

where p_i^l denotes a point in the l-th scale down-sampled point cloud $\tilde{\mathcal{P}}^l$, p_j^{l-1} and F_j^{l-1} represents the points from the neighbor set $\mathcal{N}^{l-1}(p_i^l)$ of p_i^l in the $l-1$-th scale and the corresponding extracted feature, respectively, and the symbols $MaxPool$ and MLP stand for the operator of max-pooling and the feature extractor MLP, respectively.

Subsequently, the learned feature maps in the lower scales are interpolated up to align the features in the original scale by weighting the features of three nearest neighbors, which can be formalized as

$$\begin{aligned} \tilde{F}^l &= TriInter(\tilde{\mathcal{P}}_c, \tilde{\mathcal{P}}^l, F^l), \\ \tilde{F}_i^l &= \frac{\sum_{j=1}^{3} w_i^j \cdot F_j^l}{\sum_{j=1}^{3} w_i^j}, p_i \in \tilde{\mathcal{P}}_c / \tilde{\mathcal{P}}^l, \\ w_i^j &= \frac{1}{\|p_i - p_j\|_2}, \end{aligned} \tag{10}$$

where $TriInter$ means the triple nearest neighbor interpolation, p_i denotes the point that appear in the original scale point cloud $\tilde{\mathcal{P}}_c$ but not in the current l scale point cloud $\tilde{\mathcal{P}}^l$, p_j represent the j-th nearest neighbor of p_i in $\tilde{\mathcal{P}}^l$, and F_j^{l-1} stands for the feature of the point p_j in the l-th scale.

These aligned features are separately processed by a 3-layer convolution with a kernel size of 1×1. To promote the fusion of

features, features in the near scales are concatenated and fed into an MLP to reduce the feature dimensions by half. These two-branch intermediate features are further concatenated for MLPs to predict points with the dimension size of $(G, 3\times\gamma)$, and these reconstructed points are finally reshaped to an up-sampled high-precision point cloud with the dimension size of $(G \times \gamma, 3)$. This process can be formalized as

$$\begin{aligned}\hat{\mathcal{P}} =& Reshape\Big(MLPs\Big(Concat\big(MLP\big(Concat(Convs(\tilde{\mathcal{P}}_c),\\ & Convs(\tilde{F}^1))\big), MLP\big(Concat(Convs(\tilde{F}^1), Convs(\tilde{F}^2))\big)\big)\Big)\Big).\end{aligned} \tag{11}$$

To train the Up-Net, the repulsion loss [44] is used to ensure the generated point cloud more evenly, which is defined as

$$\mathcal{L}_{REP}(\hat{\mathcal{P}}) = \sum_{p_i\in\hat{\mathcal{P}}}\sum_{j=1}^{K}\eta(\|p_i - p_j\|_2)\omega(\|p_i - p_j\|_2), \tag{12}$$

where K is the number of neighbors, $\eta(d) = -d$ denotes the repulsion term to penalize the affinity between neighboring points, and $\omega(d) = e^{-d^2}$ represents the fast-decaying weight function for the repulsion term.

Moreover, Earth Mover's distance (EMD) loss is introduced to preserve the alignment of point clouds in shape and density distribution by forcing point-to-point matching, which is defined as

$$\mathcal{L}_{EMD}(\hat{\mathcal{P}}, \mathcal{P}_G) = \sum_{p_i\in\hat{\mathcal{P}}}\min_{p_j\in\mathcal{P}_G}\left\|p_i - p_j\right\|_2, \tag{13}$$

where $\mathcal{P}_G$ denotes the ground truth high-precision point cloud. Thus, the overall loss to optimize the Up-Net can be formalized as

$$\mathcal{L}_{Up} = \mathcal{L}_{REP} + \mathcal{L}_{EMD}. \tag{14}$$

3.4 Training and Inference

Training. The proposed Down-Net and Up-Net are trained separately. Firstly, the Down-Net is trained on noisy patches generated by the Noise-Gen module using the loss $\mathcal{L}_{Down}$ in 8. The Down-Net is then frozen and employed to convert raw point clouds into center prediction results. These intermediate representations are subsequently used to optimize Up-Net with the loss $\mathcal{L}_{Up}$ in 14.

Inference. During the testing phase, the testing point cloud $\mathcal{P}_{in}$ after noise injection by Noise-Gen is reconstructed into an anomaly-free point cloud $\mathcal{P}_{out}$ using the trained Down-Net and Up-Net. Although the precision of the output point cloud P_{out} is high, the number of its points may not exactly match the raw point cloud. Thus, the nearest neighbor distance between two point clouds is calculated as the point anomaly score:

$$s_i = \min_{p_j\in\mathcal{P}_{out}}\left\|p_i - p_j\right\|_2, p_i \in \mathcal{P}_{in}. \tag{15}$$

To facilitate pixel-level anomaly detection, the anomaly score is normalized using a weighting method [20], which is denoted as:

$$\tilde{s}_i = \left(1 - \frac{\exp\left\|p_i - p_j\right\|_2}{\sum_{p_j\in\mathcal{N}_3(p_i,\mathcal{P}_{out})}\exp\left\|p_i - p_j\right\|_2}\right)s_i, \tag{16}$$

where $p_j \in \mathcal{N}_3(p_i, \mathcal{P}_{out})$ denotes the 3 nearest points of p_i in P_{out}. Further, with the maximum value of the point-level anomaly scores, the object-level score S is calculated as

$$S = \max_{p_i\in\mathcal{P}_{in}}(\tilde{s}_i) \tag{17}$$

4 Experiments

4.1 Implementation

Datasets. We conducted extensive experiments on Anomaly-Shape Net [1] and Real3D-AD [8]. (1) The Anomaly-ShapeNet has over 1,600 positive and negative samples from 40 categories, leading to a more challenging setting due to the large number of categories. (2) The Real3D-AD consists of 1,254 large-scale high-resolution point cloud samples from 12 categories, with the training set for each category only containing 4 normal samples.

Baselines. We selected BTF [3], PatchCore [20], M3DM [5], CPMF [32], Reg 3D-AD [8], IMRNet [1], R3D-AD [2], and ISMP [10] for comparison. The results of these methods were obtained by executing publicly available code or by referring to their papers.

Evaluation Metrics. We adopted Object-Level Area Under the Receiver Operator Curve (O-AUROC, ↑) and Object-Level Area Under the Per-Region-Overlap (O-AUPR, ↑) to evaluate object-level anomaly detection performance, and Point-Level Area Under the Receiver Operator Curve (P-AUROC, ↑) and Point-Level Area Under the Per-Region-Overlap (P-AUPR, ↑) to evaluate pixel-level anomaly segmentation precision. To avoid a small number of categories dominating the average performance, the average ranking (↓) was used for evaluation. In addition, Frames Per Second (FPS, ↑) was introduced to compare the inference speed of different methods.

Experimental Details. Experiments were carried out on a machine equipped with two L20 (48GB) GPUs to promote stable training with a larger batch size and evaluated on only an RTX3090 (24GB) GPU. We pre-trained the proposed Down-Net on the ShapeNetPart dataset [45] like PointMAE [29] and trained Up-Net on a subset of the Visionair repository following the setup of PU-Net [46]. The parameters α and β in Noisy-Gen were set to 0.08 and 0.15, respectively. The number of groups G and the number of points for each group K were set to 8192 and 640 for Down-Net, and the up-sampling rate γ was set to 8 for Up-Net. Three independent runs of experiments were conducted, and their performance was averaged to obtain convincing and representative results.

4.2 Main Results

Results on Anomaly-ShapeNet. Tables 1 and 3 present the detection and segmentation results of different methods on Anomaly-ShapeNet, respectively. Our method achieved an average O-AUROC and P-AUROC of 0.797 and 0.712, outperforming the second-best methods by 2.1% and 3.8%, respectively. Notably, our method obtained the best average ranking on both O-AUROC and P-AUROC, demonstrating the superior generalization on different categories.

Results on Real3D-AD. Tables 2 and 4 report the performance of different methods on Real3D-AD dataset. Our method obtained an impressive O-AUROC and P-AUROC of 0.795 and 0.847, respectively, surpassing the best baseline by 3.8% and 0.9%, respectively. Moreover, our approach yielded the best average rankings on both metrics, indicating its effectiveness for high-resolution point clouds.

Existing methods face challenges in precisely reconstructing the point cloud due to the significant difference between the training

Table 1: O-AUROC performance of different methods on Real3D-AD across 40 categories, where best and second-place results are highlighted in red and blue, respectively.

O-AUROC														
Method	ashtray0	bag0	bottle0	bottle1	bottle3	bowl0	bowl1	bowl2	bowl3	bowl4	bowl5	bucket0	bucket1	cap0
BTF(Raw) (CVPR23')	0.578	0.410	0.597	0.510	0.568	0.564	0.264	0.525	0.385	0.664	0.417	0.617	0.321	0.668
BTF(FPFH) (CVPR23')	0.420	0.546	0.344	0.546	0.322	0.509	0.668	0.510	0.490	0.609	0.699	0.401	0.633	0.618
M3DM (CVPR23')	0.577	0.537	0.574	0.637	0.541	0.634	0.663	0.684	0.617	0.464	0.409	0.309	0.501	0.557
PatchCore(FPFH) (CVPR22')	0.587	0.571	0.604	0.667	0.572	0.504	0.639	0.615	0.537	0.494	0.558	0.469	0.551	0.580
PatchCore(PointMAE) (CVPR22')	0.591	0.601	0.513	0.601	0.650	0.523	0.629	0.458	0.579	0.501	0.593	0.593	0.561	0.589
CPMF (PR24')	0.353	0.643	0.520	0.482	0.405	0.783	0.639	0.625	0.658	0.683	0.685	0.482	0.601	0.601
Reg3D-AD (NeurIPS23')	0.597	0.706	0.486	0.695	0.525	0.671	0.525	0.490	0.348	0.663	0.593	0.610	0.752	0.693
IMRNet (CVPR24')	0.671	0.660	0.552	0.700	0.640	0.681	0.702	0.685	0.599	0.676	0.710	0.580	0.771	0.737
R3D-AD (ECCV24')	0.833	0.720	0.733	0.737	0.781	0.819	0.778	0.741	0.767	0.744	0.656	0.683	0.756	0.822
DU-Net (Ours)	0.867	0.605	0.838	0.871	0.827	0.844	0.869	0.952	0.839	0.859	0.776	0.838	0.822	0.859

Method	cap3	cap4	cap5	cup0	cup1	eraser0	headset0	headset1	helmet0	helmet1	helmet2	helmet3	jar0	micro.
BTF(Raw)(CVPR23')	0.527	0.468	0.373	0.403	0.521	0.525	0.378	0.515	0.553	0.349	0.602	0.526	0.420	0.563
BTF(FPFH) (CVPR23')	0.522	0.520	0.586	0.586	0.610	0.719	0.520	0.490	0.571	0.719	0.542	0.444	0.424	0.671
M3DM(CVPR23')	0.423	0.777	0.639	0.539	0.556	0.627	0.577	0.617	0.526	0.427	0.623	0.374	0.441	0.357
PatchCore(FPFH)(CVPR22')	0.453	0.757	0.790	0.600	0.586	0.657	0.583	0.637	0.546	0.484	0.425	0.404	0.472	0.388
PatchCore(PointMAE) (CVPR22')	0.476	0.727	0.538	0.610	0.556	0.677	0.591	0.627	0.556	0.552	0.447	0.424	0.483	0.488
CPMF(PR24')	0.551	0.553	0.697	0.497	0.499	0.689	0.643	0.458	0.555	0.589	0.462	0.520	0.610	0.509
Reg3D-AD(NeurIPS23')	0.725	0.643	0.467	0.510	0.538	0.343	0.537	0.610	0.600	0.381	0.614	0.367	0.592	0.414
IMRNet(CVPR24')	0.775	0.652	0.652	0.643	0.757	0.548	0.720	0.676	0.597	0.600	0.641	0.573	0.780	0.755
R3D-AD(ECCV24')	0.730	0.681	0.670	0.776	0.757	0.890	0.738	0.795	0.757	0.720	0.633	0.707	0.838	0.762
DU-Net (Ours)	0.775	0.736	0.739	0.869	0.776	0.644	0.741	0.961	0.743	0.860	0.841	0.859	0.744	0.837

Method	shelf0	tap0	tap1	vase0	vase1	vase2	vase3	vase4	vase5	vase7	vase8	vase9	Average	Ranking
BTF(Raw)(CVPR23')	0.164	0.525	0.573	0.531	0.549	0.410	0.717	0.425	0.585	0.448	0.424	0.564	0.493	7.700
BTF(FPFH) (CVPR23')	0.609	0.560	0.546	0.342	0.219	0.546	0.699	0.510	0.409	0.518	0.668	0.268	0.528	7.025
M3DM(CVPR23')	0.564	0.754	0.739	0.423	0.427	0.737	0.439	0.476	0.317	0.657	0.663	0.663	0.552	6.800
PatchCore(FPFH) (CVPR22')	0.494	0.753	0.766	0.455	0.423	0.721	0.449	0.506	0.417	0.693	0.662	0.660	0.568	6.300
PatchCore(PointMAE) (CVPR22')	0.523	0.458	0.538	0.447	0.552	0.741	0.460	0.516	0.579	0.650	0.663	0.629	0.562	6.325
CPMF(PR24')	0.685	0.359	0.697	0.451	0.345	0.582	0.582	0.514	0.618	0.397	0.529	0.609	0.559	6.350
Reg3D-AD(NeurIPS23')	0.688	0.676	0.641	0.533	0.702	0.605	0.650	0.500	0.520	0.462	0.620	0.594	0.572	6.400
IMRNet(CVPR24')	0.603	0.676	0.696	0.533	0.757	0.614	0.700	0.524	0.676	0.635	0.630	0.594	0.661	3.925
R3D-AD(ECCV24')	0.696	0.736	0.900	0.788	0.729	0.752	0.742	0.630	0.757	0.771	0.721	0.718	0.749	2.150
DU-Net (Ours)	0.688	0.739	0.840	0.833	0.808	0.644	0.766	0.749	0.838	0.844	0.808	0.525	0.797	1.775

Table 2: O-AUROC performance of different methods on Real3D-AD across 12 categories, where best and second-place results are highlighted in red and blue, respectively.

O-AUROC														
Method	Airplane	Car	Candy	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Average	Ranking
BTF(Raw) (CVPR23')	0.730	0.647	0.539	0.789	0.707	0.691	0.602	0.686	0.596	0.396	0.530	0.703	0.635	6.538
BTF(FPFH) (CVPR23')	0.520	0.560	0.630	0.432	0.545	0.784	0.549	0.648	0.779	0.754	0.575	0.462	0.603	7.000
M3DM (CVPR23')	0.434	0.541	0.552	0.683	0.602	0.433	0.540	0.644	0.495	0.694	0.551	0.450	0.552	9.000
PatchCore(FPFH) (CVPR22')	0.882	0.590	0.541	0.837	0.574	0.546	0.675	0.370	0.505	0.589	0.441	0.565	0.593	7.692
PatchCore(PointMAE) (CVPR22')	0.726	0.498	0.663	0.827	0.783	0.489	0.630	0.374	0.539	0.501	0.519	0.585	0.594	7.769
CPMF (PR24')	0.701	0.551	0.552	0.504	0.523	0.582	0.558	0.589	0.729	0.653	0.700	0.390	0.586	8.000
Reg3D-AD (NeurIPS23')	0.716	0.697	0.685	0.852	0.900	0.584	0.915	0.417	0.762	0.583	0.506	0.827	0.704	5.231
IMRNet (CVPR24')	0.762	0.711	0.755	0.780	0.905	0.517	0.880	0.674	0.604	0.665	0.674	0.774	0.725	4.385
R3D-AD (ECCV24')	0.772	0.696	0.713	0.714	0.685	0.909	0.692	0.665	0.720	0.840	0.701	0.703	0.734	4.000
ISMP (AAAI25')	0.858	0.731	0.852	0.714	0.948	0.712	0.945	0.468	0.729	0.623	0.660	0.842	0.757	3.462
DU-Net (Ours)	0.718	0.738	0.856	0.696	0.824	0.844	0.908	0.733	0.814	0.822	0.755	0.834	0.795	2.615

and test data. Our method uses the Down-Net to keep the group centers for main structure consistency and the Up-Net to reconstruct high-precision point clouds by fusing multi-scale features, thereby leading to better detection and segmentation performance.

4.3 Ablation Studies

Our method consists of three key components: the **Noise-Gen**, the **Down-Net** with the losses $\mathcal{L}_{MSE}$, $\mathcal{L}_{COS}$, and $\mathcal{L}_{CD}$, and the **Up-Net** with the losses $\mathcal{L}_{REP}$ and $\mathcal{L}_{EMD}$. To evaluate the importance of these components, we conducted ablation studies on Anomaly-ShapeNet across 40 classes, and the results are shown in Table 5, where M_i denotes the method without the corresponding loss, $M_{w/o}$ represents the method without the corresponding component, and $M_{replace\ D}$ stands for the method by replacing the Down-Net module with FPS down-sampling.

The losses for Down-Net. For Down-Net, training without the reconstruction distance loss $\mathcal{L}_{MSE}$ (M_1) resulted in a performance degradation of 11.7%, 7.0%, 12.0%, and 15.2% in O-AUROC, P-AUROC, O-AUPR, and P-AUPR, respectively. Removing the orientation constraint loss $\mathcal{L}_{COS}$ (M_2) led to a performance reduction of 3.1%, 1.0%, 5.8%, and 6.2% in four metrics, respectively. While the Down-Net without the loss $\mathcal{L}_{CD}$ (M_3) also became worse, causing a decrease of 3.4%, 3.1%, 6.0%, and 9.7% in different metrics, respectively. These results clearly demonstrate the significance of different

Table 3: P-AUROC performance of different methods on Anomaly-ShapeNet across 40 categories, where best and second-place results are highlighted in red and blue, respectively.

P-AUROC														
Method	**ashtray0**	**bag0**	**bottle0**	**bottle1**	**bottle3**	**bowl0**	**bowl1**	**bowl2**	**bowl3**	**bowl4**	**bowl5**	**bucket0**	**bucket1**	**cap0**
BTF(Raw)(CVPR23')	0.512	0.430	0.551	0.491	0.720	0.524	0.464	0.426	0.685	0.563	0.517	0.617	0.686	0.524
BTF(FPFH)(CVPR23')	0.624	0.746	0.641	0.549	0.622	0.710	0.768	0.518	0.590	0.679	0.699	0.401	0.633	0.730
M3DM(CVPR23')	0.577	0.637	0.663	0.637	0.532	0.658	0.663	0.694	0.657	0.624	0.489	0.698	0.699	0.531
PatchCore(FPFH)(CVPR22')	0.597	0.574	0.654	0.687	0.512	0.524	0.531	0.625	0.327	0.720	0.358	0.459	0.571	0.472
PatchCore(PointMAE)(CVPR22')	0.495	0.674	0.553	0.606	0.653	0.527	0.524	0.515	0.581	0.501	0.562	0.586	0.574	0.544
CPMF(PR24')	0.615	0.655	0.521	0.571	0.435	0.745	0.488	0.635	0.641	0.683	0.684	0.486	0.601	0.601
Reg3D-AD(NeurIPS23')	0.698	0.715	0.886	0.696	0.525	0.775	0.615	0.593	0.654	0.800	0.691	0.619	0.752	0.632
IMRNet(CVPR24')	0.671	0.668	0.556	0.702	0.641	0.781	0.705	0.684	0.599	0.576	0.715	0.585	0.774	0.715
ISMP(AAAI25')	0.865	0.734	0.722	0.869	0.740	0.762	0.702	0.706	0.851	0.753	0.733	0.545	0.683	0.672
DU-Net (Ours)	0.612	0.628	0.749	0.822	0.641	0.769	0.735	0.617	0.574	0.812	0.744	0.738	0.754	0.701

Method	**cap3**	**cap4**	**cap5**	**cup0**	**cup1**	**eraser0**	**headset0**	**headset1**	**helmet0**	**helmet1**	**helmet2**	**helmet3**	**jar0**	**micro.**
BTF(Raw)(CVPR23')	0.687	0.469	0.373	0.632	0.561	0.637	0.578	0.475	0.504	0.449	0.605	0.700	0.423	0.583
BTF(FPFH)(CVPR23')	0.658	0.524	0.586	0.790	0.619	0.719	0.620	0.591	0.575	0.749	0.643	0.724	0.427	0.675
M3DM(CVPR23')	0.605	0.718	0.655	0.715	0.556	0.710	0.581	0.585	0.599	0.427	0.623	0.655	0.541	0.358
PatchCore(FPFH)(CVPR22')	0.653	0.595	0.795	0.655	0.596	0.810	0.583	0.464	0.548	0.489	0.455	0.737	0.478	0.488
PatchCore(PointMAE)(CVPR22')	0.488	0.725	0.545	0.510	0.856	0.378	0.575	0.423	0.580	0.562	0.651	0.615	0.487	0.886
CPMF(PR24')	0.551	0.553	0.551	0.497	0.509	0.689	0.699	0.458	0.555	0.542	0.515	0.520	0.611	0.545
Reg3D-AD(NeurIPS23')	0.718	0.815	0.467	0.685	0.698	0.755	0.580	0.626	0.600	0.624	0.825	0.620	0.599	0.599
IMRNet(CVPR24')	0.706	0.753	0.742	0.643	0.688	0.548	0.705	0.476	0.598	0.604	0.644	0.663	0.765	0.742
ISMP(AAAI25')	0.775	0.661	0.770	0.552	0.851	0.524	0.472	0.843	0.615	0.603	0.568	0.522	0.661	0.600
DU-Net (Ours)	0.763	0.783	0.844	0.727	0.654	0.569	0.718	0.749	0.718	0.737	0.744	0.682	0.771	0.648

Method	**shelf0**	**tap0**	**tap1**	**vase0**	**vase1**	**vase2**	**vase3**	**vase4**	**vase5**	**vase7**	**vase8**	**vase9**	**Average**	**Mean Rank**
BTF(Raw)(CVPR23')	0.464	0.527	0.564	0.618	0.549	0.403	0.602	0.613	0.585	0.578	0.550	0.564	0.550	7.575
BTF(FPFH)(CVPR23')	0.619	0.568	0.596	0.642	0.619	0.646	0.699	0.710	0.429	0.540	0.662	0.568	0.628	5.275
M3DM(CVPR23')	0.554	0.654	0.712	0.608	0.602	0.737	0.658	0.655	0.642	0.517	0.551	0.663	0.616	5.725
PatchCore(FPFH)(CVPR22')	0.613	0.733	0.768	0.655	0.453	0.721	0.430	0.505	0.447	0.693	0.575	0.663	0.580	6.650
PatchCore(PointMAE)(CVPR22')	0.543	0.858	0.541	0.677	0.551	0.742	0.465	0.523	0.572	0.651	0.364	0.423	0.577	6.875
CPMF(PR24')	0.783	0.458	0.657	0.458	0.486	0.582	0.582	0.514	0.651	0.504	0.529	0.545	0.573	7.325
Reg3D-AD(NeurIPS23')	0.688	0.589	0.741	0.548	0.602	0.405	0.511	0.755	0.624	0.881	0.811	0.694	0.668	4.125
IMRNet(CVPR24')	0.605	0.681	0.699	0.535	0.685	0.614	0.401	0.524	0.682	0.593	0.635	0.691	0.650	4.525
ISMP(AAAI25')	0.701	0.844	0.678	0.687	0.534	0.773	0.622	0.546	0.580	0.747	0.736	0.823	0.691	3.925
DU-Net (Ours)	0.740	0.728	0.743	0.699	0.648	0.650	0.731	0.765	0.711	0.728	0.762	0.581	0.712	2.900

Table 4: P-AUROC performance of different methods on Real3D-AD across 12 categories, where best and second-place results are highlighted in red and blue, respectively.

P-AUROC														
Method	**Airplane**	**Car**	**Candy**	**Chicken**	**Diamond**	**Duck**	**Fish**	**Gemstone**	**Seahorse**	**Shell**	**Starfish**	**Toffees**	**Mean**	**Mean Rank**
BTF(Raw) (CVPR23')	0.564	0.647	0.735	0.609	0.563	0.601	0.514	0.597	0.520	0.489	0.392	0.623	0.571	7.538
BTF(FPFH) (CVPR23')	0.738	0.708	0.864	0.735	0.882	0.875	0.709	0.891	0.512	0.571	0.501	0.815	0.733	4.385
M3DM (CVPR23')	0.547	0.602	0.679	0.678	0.608	0.667	0.606	0.674	0.560	0.738	0.532	0.682	0.631	6.385
PatchCore(FPFH) (CVPR22')	0.562	0.754	0.780	0.429	0.828	0.264	0.829	0.910	0.739	0.739	0.606	0.747	0.682	5.000
PatchCore(PointMAE) (CVPR22')	0.569	0.609	0.627	0.729	0.718	0.528	0.717	0.444	0.633	0.709	0.580	0.580	0.620	6.462
Reg3D-AD (NeurIPS23')	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	4.538
R3D-AD (ECCV24')	0.594	0.557	0.593	0.620	0.555	0.635	0.573	0.668	0.562	0.578	0.608	0.568	0.592	7.077
ISMP (AAAI25')	0.753	0.836	0.907	0.798	0.926	0.876	0.886	0.857	0.813	0.839	0.641	0.895	0.836	1.769
DU-Net (Ours)	0.721	0.896	0.884	0.838	0.938	0.793	0.910	0.848	0.801	0.872	0.799	0.861	0.847	1.846

losses for Down-Net in accurately preserving anomaly-free center point clouds.

Table 5: Results of ablation experiments.

Method	M_1	M_2	M_3	M_4	M_5	$M_{w/o\ N}$	$M_{Replace\ D}$	$M_{w/o\ U}$	Ours
$\mathcal{L}_{MSE}$	×	✓	✓	✓	✓	✓	×	✓	✓
$\mathcal{L}_{COS}$	✓	×	✓	✓	✓	✓	×	✓	✓
$\mathcal{L}_{CD}$	✓	✓	×	✓	✓	✓	×	✓	✓
$\mathcal{L}_{REP}$	✓	✓	✓	×	✓	✓	✓	×	✓
$\mathcal{L}_{EMD}$	✓	✓	✓	✓	×	✓	✓	×	✓
Noise-Gen	✓	✓	✓	✓	✓	×	✓	✓	✓
O-AUROC	0.680	0.766	0.763	0.733	0.573	0.469	0.550	0.620	**0.797**
P-AUROC	0.642	0.702	0.681	0.650	0.543	0.511	0.524	0.588	**0.712**
O-AUPR	0.682	0.742	0.744	0.702	0.596	0.482	0.534	0.613	**0.802**
P-AUPR	0.092	0.182	0.147	0.104	0.048	0.013	0.028	0.080	**0.244**

The losses for Up-Net. For Up-Net, training without the repulsion constraints loss $\mathcal{L}_{REP}$ (M_4) caused a performance decrease of 6.4%, 6.2%, 10.0%, and 14.0% in O-AUROC, P-AUROC, O-AUPR, and P-AUPR, respectively. The Up-Net without the overall reconstruction constraints loss $\mathcal{L}_{EMD}$ (M_5) results in a performance reduction of 22.4%, 16.9%, 20.6%, and 19.6% in four metrics, respectively. These results indicate the importance of different losses for Up-Net to reconstruct high-precision point cloud with multi-scale features.

The effect of each module. For the components of our method, the Noise-Gen module is used to inject noise on normal patches, and the performance of our method without Noise-Gen $M_{w/o\ N}$ was dramatically declined by 32.8%, 20.1%, 32.0%, and 23.1% in O-AUROC, P-AUROC, O-AUPR, and P-AUPR, respectively. Replacing the Down-Net module with FPS down-sampling $M_{replace\ D}$

Table 6: Experimental results of the effect of parameters on Noisy-Gen.

Parm. (α/β)	(0.08/0.09)	(0.08/0.12)	(0.08/0.18)	(0.04/0.15)	(0.06/0.15)	(0.10/0.15)
O-AUROC	0.767	0.796	0.784	0.791	0.788	0.793
P-AUROC	0.710	0.707	0.703	0.711	0.708	0.710
O-AUPR	0.758	0.785	0.795	0.796	0.799	0.787
P-AUPR	0.218	0.232	0.231	0.237	0.240	0.238

Module	Uniform Noise	Salt-and-Pepper Noise	Ours (0.08, 0.15)
O-AUROC	0.572	0.642	**0.797**
P-AURC	0.708	0.542	**0.712**
O-AUPR	0.588	0.627	**0.802**
P-AUPR	0.104	0.047	**0.244**

caused a performance degradation in four metrics by 24.7%, 18.8%, 26.8%, and 21.6%, respectively. While removing the Up-Net from our method $M_{w/o\ U}$ also deteriorated performance, decreasing by 17.7%, 12.4%, 18.9%, and 16.4%, respectively. These results highlight that both components contribute to detection performance, with their combination providing the most significant performance improvements. More ablation results are reported in the *Supplementary Material*.

4.4 Parameter Sensitivity Analysis

To examine the effects of different parameters on the proposed modules, we conducted parameter sensitivity experiments on Anomaly-ShapeNet.

Parameter effects on Noisy-Gen. There are two noise parameters α and β in Noisy-Gen. As shown in Table 6, Selecting appropriate noise parameters α and β for the center and the whole Patch is beneficial for anomaly detection. When these parameters were set to $\alpha = 0.08$ and $\beta = 0.15$, the best performance was achieved. This may be attributed to that the moderate noise balances data diversity and reconstruction difficulty in generating noisy patches. In addition, when replacing Gaussian noise with uniform or salt-and-pepper noises, the performance of our method was reduced on average by 19.0%, 17.4%, 19.5%, and 16.9% in O-AUROC, P-AUROC, O-AUPR, and P-AUPR, respectively. This may be attributed to that Gaussian noise is more appropriate for diversifying data in the real world and also facilitates the model to learn a robust representation.

Table 7: Experimental results of the effect of parameters on Down-Net.

Parm.(G/K)	(1024/640)	(4096/640)	**(8192/640)**	(8192/384)	(8192/128)
O-AUROC	0.618	0.707	**0.797**	0.724	0.685
P-AUROC	0.584	0.638	**0.712**	0.69	0.659
O-AUPR	0.63	0.681	**0.802**	0.708	0.671
P-AUPR	0.029	0.133	**0.244**	0.208	0.134
FPS	3.4	3	**2.1**	2.5	2.9

Parameter effects on Down-Net. The Down-Net module involves two adjustable parameters: the number of groups G and the number of points K in each group. As shown in Table 7, the increase of G from 1024 to 8192 brought a performance improvement of 17.9%, 12.8%, 17.2%, and 21.5% in O-AUROC, P-AUROC, O-AUPR, and P-AUPR, respectively. Similarly, increasing points K for center generation from 128 to 640 enhanced the anomaly detection metrics by 11.2%, 7.4%, 12.1%, and 11.1%, respectively. To balance computational efficiency and detection accuracy, the settings $G = 8192$ and $K = 512$ were adopted in the experiments.

Table 8: Experimental results of the effect of parameters on Up-Net.

Ratio	$\gamma = 1$	$\gamma = 2$	$\gamma = 4$	$\gamma = 6$	$\gamma = 8$
O-AUROC	0.620	0.644	0.727	0.743	**0.797**
P-AUROC	0.588	0.601	0.637	0.685	**0.712**
O-AUPR	0.613	0.647	0.723	0.758	**0.802**
P-AUPR	0.080	0.114	0.138	0.188	**0.244**
FPS	4.4	3.1	2.6	2.3	**2.1**

Parameter effects on Up-Net. Point cloud reconstruction in Up-Net relies on the up-sampling rate γ. As seen in Table 8, increasing γ from 2 to 8 improved detection performance by 17.7%, 12.4%, 18.9%, and 23.6% in four metrics, with the inference efficiency FPS decreasing from 4.4 to 2.1. The reason for this may be that a larger up-sampling rate provides more points for the reconstruction of the high-resolution point cloud, allowing for better anomaly detection performance. But a high up-sampling rate leads to more computational costs. Thus, this parameter was limited to 8 in the experiments.

4.5 Robustness and Stability Analysis

Equipment instability and improper operations may result in low-quality point clouds. We considered two realistic challenges: incomplete point clouds and noisy point clouds.

Table 9: Results of robustness experiments.

Scale	1/2	1/4	1/6	1/8	Origin
O-AUROC	0.788	0.793	0.772	0.758	**0.797**
P-AUROC	0.710	0.701	0.654	0.640	**0.712**
O-AUPR	0.798	0.800	0.764	0.735	**0.802**
P-AUPR	0.238	0.232	0.210	0.183	**0.244**

(a) Incomplete point clouds.

Deviations	0.001	0.003	0.005	0.007	0.009	Origin
O-AUROC	0.796	0.780	0.769	0.751	0.713	**0.797**
P-AUROC	0.711	0.705	0.699	0.696	0.674	**0.712**
O-AUPR	0.787	0.782	0.778	0.759	0.720	**0.802**
P-AUPR	0.237	0.231	0.224	0.218	0.184	**0.244**

(b) Noisy point clouds.

Incomplete Point Clouds. An incomplete point cloud means that some points are missed during data collection, and we examined three scales: point cloud with a down-sampling rate of 1/2, 1/4, 1/6, and 1/8. As shown in Table 9a, our method maintained outstanding performance when facing incomplete point clouds. For instance, the anomaly detection performance only decreased by 3.9%, 7.2%, 6.7%, and 6.1% in a down-sampling rate of 1/8, which is still higher than the second-best comparative methods in O-AUROC, demonstrating the robustness of our method in the conditions of incomplete point cloud.

Noisy Point Clouds. We simulated noisy point clouds by injecting Gaussian noise with a standard deviation of 0.001, 0.003, 0.005, 0.007, or 0.009, respectively. As seen in Table 9b, our approach achieved good performance in noisy situations, For example, when the standard deviation was 0.009, our approach still maintained a good performance with O-AUROC, P-AUROC, O-AUPR, and P-AUPR of 71.3%, 67.4%, 72.0% and 18.4%, respectively, which are still competitive with some other methods. This may be due to that our method employs a group center-based reconstruction mechanism

and that our Down-Net has the ability to remove noise from points or patches.

5 Conclusion

High-resolution point clouds are characterized by large scale and high complexity. This paper proposes a Down-Up sampling Network (DUS-Net) to address the challenge of reconstructing high-resolution point clouds for 3D anomaly detection. To diversify the training data and provide denoising ability, we introduce a Noise Generation (Noise-Gen) module to generate noisy patches and present a Down-sampling Network (Down-Net) to predict robust and anomaly-free group center points from these corrupted patches. Upon these precise center points, we design an Upsampling Network (Up-Net) to achieve high-fidelity anomaly-free point cloud reconstruction through hierarchical aggregation of multi-scale features. Extensive experiments demonstrate that the proposed method achieves SOTA performance across different evaluation metrics while maintaining strong robustness, offering a promising solution for high-resolution industrial 3D anomaly detection. **Limitation.** Two-stage reconstruction requires additional training and cost. Exploring single-stage frameworks is worthy of further research.

Acknowledge

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62476171 and 62206122), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011367), the National Undergraduate Training Program for Innovation and Entrepreneurship (S202510590118), the Guangdong Provincial Key Laboratory (Grant No. 2023B1212060076), and the Tencent "Rhinoceros Birds" - Scientific Research Foundation for Young Teachers of Shenzhen University.

References

[1] 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," 2023.

[2] Z. Zhou, L. Wang, N. Fang, Z. Wang, L. Qiu, and S. Zhang, "R3d-ad: Reconstruction via diffusion for 3d anomaly detection," 2024.

[3] 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*, pp. 2968–2977, 2023.

[4] M. Rudolph, T. Wehrbein, B. Rosenhahn, and B. Wandt, "Asymmetric student-teacher networks for industrial anomaly detection," in *Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2023.

[5] 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*, pp. 8032–8041, 2023.

[6] K. Long, G. Xie, L. Ma, J. Liu, and Z. Lu, "Revisiting multimodal fusion for 3d anomaly detection from an architectural perspective," 2024.

[7] 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*, SCITEPRESS - Science and Technology Publications, 2022.

[8] 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," 2023.

[9] 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*, MM '24, p. 4680–4689, ACM, Oct. 2024.

[10] H. Liang, G. Xie, C. Hou, B. Wang, C. Gao, and J. Wang, "Look inside for more: Internal spatial modality perception for 3d anomaly detection," 2024.

[11] 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*, pp. 6185–6194, 2023.

[12] A. Costanzino, P. Zama Ramirez, G. Lisanti, and L. Di Stefano, "Multimodal industrial anomaly detection by crossmodal feature mapping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. CVPR.

[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] 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.

[16] T. Liu, B. Li, X. Du, B. Jiang, L. Geng, F. Wang, and Z. Zhao, "Simple and effective frequency-aware image restoration for industrial visual anomaly detection," *Advanced Engineering Informatics*, vol. 64, p. 103064, Mar. 2025.

[17] X. Tao, X. Gong, X. Zhang, S. Yan, and C. Adak, "Deep learning for unsupervised anomaly localization in industrial images: A survey," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, p. 1–21, 2022.

[18] X. Yan, H. Zhang, X. Xu, X. Hu, and P.-A. Heng, "Learning semantic context from normal samples for unsupervised anomaly detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 3110–3118, May 2021.

[19] J. Wyatt, A. Leach, S. M. Schmon, and C. G. Willcocks, "Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 650–656, June 2022.

[20] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," 2022.

[21] J. Bae, J.-H. Lee, and S. Kim, "Pni: Industrial anomaly detection using position and neighborhood information," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6350–6360, 2023.

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

[23] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[24] 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.

[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] A. Rani, D. Ortiz-Arroyo, and P. Durdevic, "Advancements in point cloud-based 3d defect classification and segmentation for industrial systems: A comprehensive survey," *Information Fusion*, vol. 112, p. 102575, Dec. 2024.

[29] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pp. 604–621, Springer, 2022.

[30] R. C. Bolles and M. A. Fischler, "A ransac-based approach to model fitting and its application to finding cylinders in range data," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, (San Francisco, CA, USA), p. 637–643, Morgan Kaufmann Publishers Inc., 1981.

[31] 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.

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

[33] Q. Zhou, J. Yan, S. He, W. Meng, and J. Chen, "Pointad: Comprehending 3d anomalies from points and pixels for zero-shot 3d anomaly detection," in *Advances in Neural Information Processing Systems* (A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, eds.), vol. 37, pp. 84866–84896, Curran Associates, Inc., 2024.

[34] Y. Wang, K.-C. Peng, and Y. R. Fu, "Towards zero-shot 3d anomaly localization," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[35] Y. Tang, Z. Guo, Z. Wang, R. Zhang, Q. Chen, J. Liu, D. Qu, Z. Wang, D. Wang, X. Li, and B. Zhao, "Exploring the potential of encoder-free architectures in 3d lmms," 2025.

[36] J. Xu, S.-Y. Lo, B. Safaei, V. M. Patel, and I. Dwivedi, "Towards zero-shot anomaly detection and reasoning with multimodal large language models," 2025.

[37] Y. Wang, K.-C. Peng, and Y. Fu, "Towards zero-shot 3d anomaly localization," 2024.
[38] 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.
[39] Y. Liu, Y. S. Hu, Y. Chen, and J. Zelek, "Splatpose+: Real-time image-based pose-agnostic 3d anomaly detection," 2024.
[40] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017.
[41] J. Ren, M. Zhang, C. Yu, and Z. Liu, "Balanced mse for imbalanced visual regression," 2022.
[42] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Density-aware chamfer distance as a comprehensive metric for point cloud completion," 2021.
[43] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," 2017.
[44] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Pu-net: Point cloud upsampling network," 2018.
[45] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," 2015.
[46] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Pu-net: Point cloud upsampling network," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.