This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Hierarchical Point-Patch Fusion with Adaptive Patch Codebook for 3D Shape Anomaly Detection

Xueyang Kang[1*†] Zizhao Li[1†] Tian Lan[2] Dong Gong[3] Kourosh Khoshelham[1]
Liangliang Nan[4‡]

[1]The University of Melbourne, Parkville, Victoria 3010, Australia
[2]Tsinghua University, Haidian District, Beijing 100871, China
[3]The University of New South Wales, Sydney, NSW 2052, Australia
[4]Delft University of Technology, Mekelweg 5, 2628 CD Delft, Netherlands

Abstract

3D shape anomaly detection is a crucial task for industrial inspection and geometric analysis. Existing deep learning approaches typically learn representations of normal shapes and identify anomalies via out-of-distribution feature detection or decoder-based reconstruction. They often fail to generalize across diverse anomaly types and scales, such as global geometric errors (e.g., planar shifts, angle misalignments), and are sensitive to noisy or incomplete local points during training. To address these limitations, we propose a hierarchical point–patch anomaly scoring network that jointly models regional part features and local point features for robust anomaly reasoning. An adaptive patchification module integrates self-supervised decomposition to capture complex structural deviations. Beyond evaluations on public benchmarks (Anomaly-ShapeNet and Real3D-AD), we release an industrial test set with real CAD models exhibiting planar, angular, and structural defects. Experiments on public and industrial datasets show superior AUC-ROC and AUC-PR performance, including over 40% point-level improvement on the new industrial anomaly type and average object-level gains of 7% on Real3D-AD and 4% on Anomaly-ShapeNet, demonstrating strong robustness and generalization. The code and test data are publicly available at: github.com/Shape-Anomaly-Codebook.git

1. Introduction

Shape anomaly detection is a fundamental industrial challenge, aiming to identify and localize structural or geometric defects in manufactured 3D products. Conventional autonomous approaches typically rely on geometric comparisons between a normal template and a test shape. A set of normal shapes is encoded into latent features as a reference, and anomalies are detected by comparing the feature distributions using Out-of-Distribution (OOD) detection [21, 23, 24, 36, 38]. Early research focused on 2D image anomaly detection, such as MVTec AD [3], where methods detect anomalies by identifying appearance differences (*e.g.*, cracks) between test images and templates [12, 13, 19].

With the emergence of 3D anomaly detection benchmarks such as Real3D-AD [17], MVTec 3D-AD [4], and Anomaly-ShapeNet [14], research has increasingly shifted toward 3D geometric domains. Recent approaches, including dual-memory branches [8] and teacher–student architectures [25], model the distribution gap between nor-

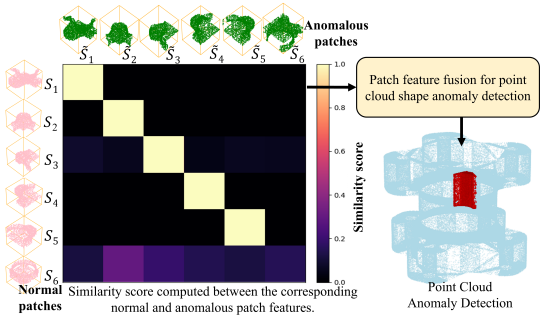


Figure 1. **Patch feature fusion for 3D shape anomaly detection in point clouds**. Normal patches $\{\mathbf{S}_j\}, j \in \{1, \ldots, 6\}$ and abnormal patches $\{\tilde{\mathbf{S}}_j\}$ are ordered by distance to the object center. The left heatmap visualizes the cosine similarity between patch features, where higher values indicate stronger similarity between normal and abnormal patches. Anomalous regions exhibit distinctive feature discrepancies in the last row, indicating ambiguous patch correspondences. These patch-level differences are fused to guide pointwise anomaly detection, with red regions in the right point cloud marking detected anomalies.

*Project leader: kangxueyang@126.com
†Zizhao contributed to baselines and backbone choice.
‡Corresponding author: Liangliang.Nan@tudelft.nl

mal and anomalous features. Multimodal methods further improve discrimination by fusing RGB images and point clouds [9, 20, 32]. Zero-shot and few-shot settings are explored by aligning point clouds with multi-view image projections [6, 40] or leveraging view synthesis [44]. Despite these advances, most methods struggle with large-scale structural anomalies and planar displacements common in industrial scenarios, as they rely primarily on local point representations. Reconstruction-based methods, such as DRAEM-A [35] and R3D-AD [41], reconstruct normal shapes and detect anomalies via reconstruction errors, but are sensitive to noise, unstable in planar or concave regions, and depend heavily on negative augmentation. Similarly, point-wise or keypoint-based approaches, including 3DKeyAD [32], geometric descriptor methods [2], and PO3AD [34], focus on per-point regression and are effective mainly for small local defects, while remaining inadequate for large structural displacements or part misalignments that are underrepresented in public benchmarks.

To address these challenges, we first construct a small industrial test dataset consisting of eight real components with anomalies such as planar displacements and large gear-angle misalignments. We then propose an **adaptive multi-scale patch-based encoding framework** for 3D shape anomaly detection. Our method extracts multi-scale patches from normal shapes and stores their features in a lightweight, position-invariant **patch feature codebook**, which is queried by pseudo-negative inputs during training to infer patch-level discrepancies. Unlike point-based methods that rely only on local geometry, this design provides multi-scale references for detecting anomalies at different scales, enabling more robust and scale-aware reasoning. Although patch-based decomposition has been explored in segmentation and completion tasks [1, 26], our approach introduces adaptive patch scales tailored to anomaly detection. During inference, anomalous shapes undergo the same multi-scale patchification, and their patch features are compared with the codebook to compute patch similarity scores. To summarize, our main contributions are threefold:

- An **adaptive patchification module** that extracts multi-scale patches from normal shapes and caches them in a lightweight, position-invariant **patch codebook**, which captures the invariant geometry of various scales.
- **A patch–point feature fusion model** that integrates patch-level and point-level features via cross-attention. A post-modulation based on patch similarity between normal and anomalous regions enables adaptive discrimination of anomalies across diverse scales and types.
- **A new industrial 3D anomaly test set** featuring realistic manufacturing defects such as gear cracks and planar displacements. Extensive experiments demonstrate superior performance with over **40%** improvement on industrial anomalies and average gains of **7%** on Real3D-AD, highlighting strong robustness over state-of-the-art baselines.

2. Related Work

Anomaly detection applications. The MVTec-AD [3] benchmark catalyzed extensive research in image anomaly detection, spawning methods that model normal distributions or synthesize pseudo anomalies [12, 19, 27, 35, 37, 38]. In 3D vision, shape anomaly detection identifies geometric deviations from learned normal shapes through *OOD detection*, where anomalous inputs fall outside the normal feature distribution. MVTec-3D-AD [4] enabled multimodal methods combining RGB and point cloud [9, 30, 31, 44], though gains over single-modality approaches remain marginal. OOD detection extends beyond industrial inspection to medical imaging [11, 43] and hyperspectral sensing [33, 43], with summary surveys available [15].

Reconstruction-based anomaly detection. A major line of anomaly detection research relies on reconstruction, where models learn to reproduce normal inputs and detect deviations [35]. These methods train models to learn geometric representations of normal shapes and reconstruct point clouds for comparison with inputs at inference. IMRNet [14], for instance, employs a masked autoencoder to reconstruct point clouds, while R3D-AD [41] leverages diffusion to recover normal distributions. Lastly, PO3AD [34] directly regresses point-wise offsets between pseudo anomalies and normal inputs.

Memory bank-based anomaly detection. Memory bank mechanisms have been extensively adopted in computer vision tasks such as semantic segmentation [1, 39] and object detection [29]. For anomaly detection, early works integrated a memory module with an autoencoder to store normal patterns [10], retrieving the closest memory feature for reconstruction, where large reconstruction errors signal anomalies. Subsequent methods extended this concept to geometric data by mapping local shape features into a memory space representing the normal feature distribution, so that anomalous inputs are naturally projected out-of-distribution. Some approaches employ hybrid representation learning across RGB and point cloud modalities [40] for anomaly detection. In contrast, others use a shared cross-modal memory [31] to fuse visual and geometric features before anomaly classification.

3. Method

We generate pseudo-abnormal shapes $\tilde{\mathbf{S}}$ from normal inputs $\mathbf{S}$ via negative augmentation [34], enabling robust anomaly discrimination. During inference, only $\tilde{\mathbf{S}}$ is provided, with patch features retrieved from a pre-cached normal codebook. Both $\mathbf{S} \in \mathbb{R}^{N\times 3}$ and $\tilde{\mathbf{S}} \in \mathbb{R}^{N\times 3}$ are divided into patches $\mathbf{S}_j$ and $\tilde{\mathbf{S}}_j$ via **Farthest Point Sampling (FPS) patchification**. The **patch feature extraction module (a)**

encodes patch centers into normal features $\mathbf{t}_k \in \mathbb{R}^d$ (template) and anomalous features $\mathbf{p}_j \in \mathbb{R}^d$. Normal features populate the **shape patch codebook (b)** for anomaly retrieval. During inference, patch correspondences between $\mathbf{p}_j$ and $\mathbf{t}_k$ are established via similarity-based querying, followed by **point-to-patch cross-attention (c)** that fuses local point features with nearest normal patch embeddings. The **part score modulation module (d)** refines fused features through gated modulation and predicts point-wise offset $\hat{\mathbf{o}} \in \mathbb{R}^{N\times 3}$, which is mapped by scoring function $\phi : \mathbb{R}^{N\times 3} \to \mathbb{R}$ to normalized anomaly scores in $[0, 1]$.

3.1. Patchification & negative augmentation

During training, given a normal shape $\mathbf{S}$, we apply **negative augmentation** to generate pseudo-anomalous shapes $\tilde{\mathbf{S}}$. Specifically, random points on $\mathbf{S}$ are selected as anomaly centers, around which local geometric deformations are introduced using Gaussian kernel-based displacements guided by the local normal directions. Positive and negative displacements simulate concave or sink-type anomalies, following the strategy of Real3D-AD [17]. To further simulate bulging deformations, the normal directions are modulated by a sine wave function, creating alternating signs of deformation. Additionally, random holes and planar cut-offs using cubic or cylindrical masks are applied to mimic missing or displaced surface regions. This augmentation strategy generates diverse, realistic structural defects that resemble those observed in real test cases.

We then adopt an **adaptive patchification** process that extracts patches at three scales (*e.g., 8, 32, 64*), providing multi-scale references that help the model distinguish anomalies of varying sizes and improve robustness to noisy points. Specifically, for each level $l \in \{1, 2, 3\}$, patches of size p_l are sampled from $\mathbf{S}$, where $\mathbf{p}^1 < \mathbf{p}^2 < \mathbf{p}^3$ correspond to fine, medium, and coarse resolutions, respectively. These multi-scale patches capture geometric details across varying granularities, enabling the network to model local fine structures and broader contextual cues jointly.

3.2. Adaptive patch feature extraction

For each patch at level $l \in \{1, 2, 3\}$, denoted as $\tilde{\mathbf{S}}_j^{(l)}$ containing points $\{\mathbf{x}_i^{j,(l)} \in \mathbb{R}^3 \mid i \in \mathcal{N}_j^{(l)}\}$, where $\mathcal{N}_j^{(l)}$ denotes the point set of patch $\tilde{\mathbf{S}}_j^{(l)}$, point-wise features are extracted using a pre-trained 3D UNet encoder. Each point $\mathbf{x}_i^{j,(l)}$ is processed to obtain its feature representation $\mathbf{z}_i^{(l)}$. The corresponding patch feature $\mathbf{p}_j^{(l)}$ is computed as the mean of all point coordinates relative to the patch centroid and processed by the pre-trained encoder f:

$$\mathbf{p}_j^{(l)} = f\left(\frac{1}{|\mathcal{N}_j^{(l)}|}\sum_{i\in\mathcal{N}_j^{(l)}}(\mathbf{x}_i^{j,(l)} - \mathbf{c}_j^{(l)})\right). \tag{1}$$

Note that the patch center $\bar{\mathbf{x}}_i^{(l)}$ for the query differs from the geometric sphere cluster center $\mathbf{c}_j^{(l)}$ used for encoder querying. Specifically, $\mathbf{p}_j^{(l)}$ denotes the centroid feature computed from the actual point distribution within the patch, whereas the center feature corresponds to the geometric sphere center. For clarity, we omit the superscript j and level l in subsequent notations; all point indices i implicitly correspond to their associated patch j and scale level l. The same process applies to template patch feature $\mathbf{t}_k^{()}$.

3.3. Shape patch feature codebook

During training, the extracted patch features at scale l, $\{\mathbf{t}_k^{(l)}\}, k \in \mathcal{N}_k^{(l)}$, from the normal shape input $\mathbf{S}$ are stored in a multi-scale **shape patch feature codebook**, where the hash key of each patch position serves as the key, and the corresponding patch features form the value set. The patch features are ordered by their distance from the patch center to the object centroid. To reduce redundancy and limit memory usage, the codebook patch features are extracted from the relative point distribution within each patch, ensuring translation invariance. When symmetry is present, as shown in the input of Fig. 2, identical corner regions share the same patch feature, which is stored once and referenced by position for reuse during repeated retrieval. This process is repeated independently for each scale $l \in \{1, 2, 3\}$ to build codebooks for fine, medium, and coarse resolutions.

During inference, the patch features of a test input at scale l, $\{\mathbf{p}_j^{(l)}\}$, serve as queries to retrieve the most similar entries $\{\mathbf{t}_j^{(l)}\}$ from the corresponding scale codebook using cosine similarity. The overall similarity score at each scale is computed by summing the feature similarities along the diagonal positions:

$$\alpha^{(l)} = \sum_{j=1}^{\mathcal{N}_{\text{Patch}}^{(l)}} (\mathbf{p}_j^{(l)} \cdot \mathbf{t}_j^{(l)}), \tag{2}$$

where a higher $\alpha^{(l)}$ indicates stronger correspondence between the test and normal patches at scale l. The scale l^* with the maximal similarity sum is selected through $\text{argmax}_{l\in 1,2,3}\alpha^{(l)}$, and the retrieved patch features at scale l^* are used to guide anomaly-aware feature modulation for final prediction. This multi-scale retrieval ensures that both local and global patch correspondences are considered, enhancing robustness to geometric scale variations.

The codebook update follows a thresholded similarity scheme. Each normalized patch feature is compared to existing entries using cosine similarity; features exceeding a threshold τ are merged via count-weighted averaging to suppress redundancy, while dissimilar features create new entries to preserve geometric diversity. Spatial hash keys from 3D positions are stored to maintain coverage across

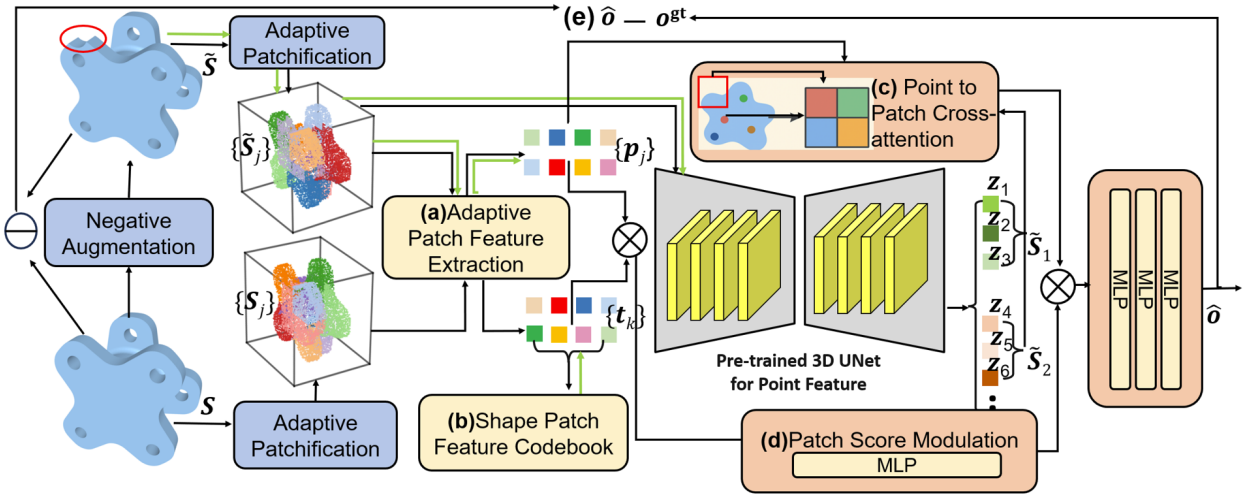


Figure 2. **Overview of the proposed shape anomaly detection framework**. The pipeline processes a normal shape $\mathbf{S}$ through several modules: **Patchification** segments the input into patches $\{\mathbf{S}_j\}$, while **Negative Augmentation** generates pseudo anomalous shapes $\tilde{\mathbf{S}}$ that are also patchified during training (red regions indicate anomalies); (a) **Adaptive Patch Feature Extraction** computes features for each patch by averaging its points and querying a pre-trained UNet to obtain patch features $\{\mathbf{p}_j\}$; (b) **Template Codebook** stores a dictionary of normal template patch features $\{\mathbf{t}_k\}$, from which each patch retrieves its closest matching patch; (c) **Point to Patch Cross-attention** fuses the template features $\{\mathbf{t}_k\}$ with point-level features extracted from the pre-trained UNet encoder through multi-head attention mechanism; (d) **Patch Score Modulation** modulates the cross-attention outputs using similarity scores between normal and abnormal patch features $\{\mathbf{t}_k\}, \{\mathbf{p}_j\}$, with the modulated features fed into MLP regression layers to predict the anomaly score δ_{pred}. During training, (e) the loss is computed as $\hat{\mathbf{o}} - \mathbf{o}_{gt}$, where $\mathbf{o}_{gt}$ represents the ground-truth offset direction derived from the difference $\mathbf{S} - \tilde{\mathbf{S}}$. During inference (indicated by green arrows), only the test shape is input for anomaly detection.

locations. We provide the pseudo code for the algorithm implementation in Algorithm 1.

Algorithm 1 Adaptive Patch Codebook Update

Require: Patch feature $\mathbf{t}_j^{(l)} \in \mathbb{R}^{32}$, hash keys $\mathcal{H}_j^{(l)} = H^{(l)}(\mathbf{p}_j = (x_j, y_j, z_j))$, codebook $\mathcal{C}^{(l)}$, threshold τ, Merging times n_i
Ensure: Updated codebook $\mathcal{C}^{(l)}$

```
Normalize: t_j^(l) ← t_j^(l) / ||t_j^(l)||_2
for each (c_i^(l), H_i^(l)) ∈ C^(l) do ▷ Iterate over codebook
    s_i ← (t_j^(l))^⊤ c_i^(l)  ▷ Cosine similarity
    if s_i ≥ (τ = 0.85) then  ▷ Merge similar features
        c_i^(l) ← (n_i c_i^(l) + s_i t_j^(l)) / (n_i + s_i),  ▷ Count-weighted merge
        H_i^(l) ← H_i^(l) ∪ {H^(l)(p_j)}  ▷ Add position hash
        return C^(l)
    end if
end for
C^(l) ← C^(l) ∪ {(t_j^(l), {H^(l)(p_j)})}  ▷ Add new entry
return C^(l)
```

3.4. Point-to-patch cross attention

Each point coordinate $\mathbf{x}_i$ from the pre-trained U-Net encoder is associated with a point feature $\mathbf{z}_i$. During inference, the point features $\{\mathbf{z}_i\}$ are mapped to query tokens $\{\mathbf{q}_i\}$, while the retrieved patch codebook features $\{\mathbf{t}_k\}$ serve as key-value tokens $\{\mathbf{k}_i, \mathbf{v}_i\}$ in a RoPE-based multi-head cross-attention module.

Rotary position embedding. To incorporate relative spatial relationships between points and their corresponding patches, we adopt Rotary Position Embedding (RoPE) [28]. RoPE encodes relative positional information by rotating feature embedding according to the angular offset between a point $\mathbf{x}_i$ and its patch center $\bar{\mathbf{x}}i$:

$$\text{RoPE}(\mathbf{x}_i, \bar{\mathbf{x}}_i) = \mathbf{R}^d_{\Theta,i}\,(\mathbf{x}_i - \bar{\mathbf{x}}_i). \tag{3}$$

where $\mathbf{R}^d_{\Theta,i}$ is a rotation matrix applied over the d-D embedding space and parameterized by frequency Θ.

RoPE-enhanced cross attention. Given the RoPE-transformed embeddings, cross-attention between point and patch tokens is then computed as,

$$\text{Attn}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \sum_{i=1}^{N} \alpha_i\, \mathbf{v}_i = \hat{\mathbf{z}}_i, \tag{4}$$

$$\alpha_i = \frac{\sum_{n=1}^{N} \left(\mathbf{R}^d_{\Theta,n}\,\phi(\mathbf{q}_n)\right)^\top \left(\mathbf{R}^d_{\Theta,n}\,\varphi(\mathbf{k}_n)\right) \mathbf{v}_n}{\sum_{n=1}^{N} \phi(\mathbf{q}_n)^\top \varphi(\mathbf{k}_n)}, \tag{5}$$

where $\mathbf{q}_n$, $\mathbf{k}_n$, and $\mathbf{v}_n$ denote the query, key, and value, respectively. $\phi(\cdot)$ and $\varphi(\cdot)$ are non-negative feature mappings, typically $\phi(x) = \varphi(x) = \text{elu}(x) + 1$. Multi-head concatenation followed by linear projection yields updated point tokens $\{\hat{\mathbf{z}}_i\}$, enabling patch-guided, geometry-aware feature fusion for anomaly detection. This cross-attention

mechanism embeds patch-level priors into point-level features. By attending to retrieved normal patch embeddings, each point aligns with its normal patch geometry, improving detection of both fine-grained local defects and large structural misalignments.

3.5. Patch score modulation

Given the normalized patch features $\{\mathbf{p}_j\}$ of an anomalous shape and the patch features $\{\mathbf{t}_k\}$ of a normal shape, the patch-level discrepancy is first computed as $\Delta\mathbf{f}_{kj} = (1 - \mathbf{t}_k \cdot \mathbf{p}_j)$ between the matched normal patch k and the corresponding anomaly patch j. A gating function $\sigma_i(\cdot)$ estimates the anomaly likelihood $\rho_i = \sigma_i(\text{MLP}_{\text{gate}}(\Delta\mathbf{f}_{kj}))$ of ith feature, while a modulation network $\mathcal{M}_{\text{mod}}(\cdot)$ predicts the feature-wise scaling and shifting parameters $(\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i) = \mathcal{M}_{\text{mod}}(\Delta\mathbf{f}_{kj})$. The final modulated feature is defined as:

$$\mathbf{z}'_i = \rho_i \odot (\boldsymbol{\gamma}_i \odot \hat{\mathbf{z}}_i + \boldsymbol{\beta}_i), \tag{6}$$

where $\hat{\mathbf{z}}_i$ is the cross-attended feature in Eq. (4), $\odot$ represents element-wise multiplication, and $\mathbf{z}'_i$ is the modulated point feature corresponding to the i-th point within the j-th patch after RoPE-enhanced cross-attention. The gate $\rho_i \in [0, 1]$ adaptively weights each patch contribution to the anomaly score based on the inverse feature similarity in Eq. (2).

Finally, the refined point features $\mathbf{z}'_i$ are concatenated with the attended point features $\hat{\mathbf{z}}_i$ and passed through a multilayer perceptron (MLP) for residual prediction of the anomaly offset vector $\hat{\mathbf{o}}_i$:

$$\hat{\mathbf{o}}_i = \text{MLP}\left(Concat[\mathbf{z}'_i, \hat{\mathbf{z}}_i]\right) + \hat{\mathbf{z}}_i, \tag{7}$$

where $\hat{\mathbf{o}}_i$ represents the predicted anomaly offset vector for each point. This gated modulation mechanism adaptively emphasizes discriminative patch–point relationships to detect precise shape anomalies of varying sizes.

3.6. Loss function

For each point $\mathbf{x}_i$ in an abnormal shape, let $\mathbf{o}_i^{\text{gt}} \in \mathbb{R}^3$ denote the ground-truth offset vector from the abnormal point to its corresponding normal counterpart, and $\hat{\mathbf{o}}_i$ be the predicted offset. The network jointly predicts (a) the point-wise anomaly offset $\hat{\mathbf{o}}_i$, and (b) a direction mask $\hat{m}_i$ indicating the sign of deviation. The overall anomaly loss combines three complementary terms:

$$\mathcal{L}_{\text{anomaly}} = \mathcal{L}_{\text{dist}} + \lambda_{\text{sim}}\mathcal{L}_{\text{sim}} + \lambda_{\text{BCE}}\mathcal{L}_{\text{BCE}}, \tag{8}$$

where λ_{sim} and λ_{BCE} balance the offset direction and mask label supervision. The offset regression loss minimizes the mean absolute distance between predicted and ground-truth offset vectors:

$$\mathcal{L}_{\text{dist}} = \frac{1}{N}\sum_{i=1}^{N} \left\|\hat{\mathbf{o}}_i - \mathbf{o}_i^{\text{gt}}\right\|_1. \tag{9}$$

To encourage alignment between the predicted and true offset directions, a cosine similarity loss is used:

$$\mathcal{L}_{\text{sim}} = -\frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\left(1 + \frac{\hat{\mathbf{o}}_i \cdot \mathbf{o}_i^{\text{gt}}}{\|\hat{\mathbf{o}}_i\|_2 \, \|\mathbf{o}_i^{\text{gt}}\|_2 + \epsilon}\right), \tag{10}$$

where ϵ is a small constant (10^{-6}) for numerical stability. Finally, a binary cross-entropy loss supervises the predicted anomaly sign mask $\hat{m}_i$ against its ground-truth label m_i^{gt}:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N} \left[m_i^{\text{gt}} \log(\hat{m}_i) + (1 - m_i^{\text{gt}}) \log(1 - \hat{m}_i)\right]. \tag{11}$$

This combined loss jointly constrains the magnitude, direction, and offset sign consistency of anomaly predictions, enabling stable training and accurate point-wise shape anomaly localization. During inference, the predicted offset magnitude $\hat{\mathbf{o}}_i$ is normalized into the anomaly score $\hat{\delta}_i$ using the same $\mathcal{L}_1$ normalization strategy as in PO3AD [34], yielding anomaly score $\hat{\delta}_i$ within $[0, 1]$. Final anomaly scores are calculated solely for points designated as valid by the mask.

4. Experiments & Results

4.1. Experimental setup

Datasets & preprocessing. We primarily use two benchmark 3D point-cloud anomaly-detection datasets for evaluation, including **Real3D-AD** [17] and **Anomaly ShapeNet** [14]. The Anomaly ShapeNet dataset contains 40 categories with a total of 1,600 samples, including both normal and anomalous shapes. Real3D-AD includes 12 categories, each with 4 normal template samples and 100 test samples containing anomalies. Additionally, we introduce **a new industrial evaluation dataset** comprising 8 normal samples and 8 corresponding test samples with angular misalignment and planar displacements. Since Real3D-AD test scans are half-sided, while the training data are full 360° scans, we adopt the same half-sided configuration during codebook construction for consistency.

For preprocessing, negative augmentation is applied to two geometric scales (0.01 and 0.1), while angular displacement uses intersection cut-off ratios between 0.1 and 0.5. All inputs are normalized into canonical space and voxelized into 256^3 grids. For Anomaly ShapeNet and Real3D-AD, multi-scale patching is performed at three scales with patch numbers $32, 64, 192$ and corresponding patch sizes $64, 32, 8$. For our own test set, we adopt inverse patch scaling with patch numbers $8, 32, 64$ and corresponding patch sizes $192, 64, 32$ to emphasize large structural variations.

Baselines. We follow recent state-of-the-art learning-based baselines, including PO3AD [34], R3D-AD [41], and

ISMP [16], using their official implementations and default configurations. For qualitative comparison, we visualize the results using the provided training code for these methods. Since pre-trained weights are unavailable for several categories, only PO3AD is evaluated on our newly created test samples, as other models fail to generalize to unseen categories. Evaluation metrics include the *Area Under the Receiver Operating Characteristic Curve (AUC-ROC)* and the *Area Under the Precision-Recall Curve (AUC-PR)*.

Training & evaluation. All categories are trained for 1500 epochs using the Adam optimizer with a learning rate of 1×10^{-3}. The weighting coefficients λ_{sim} and λ_{BCE} in Eq. (8) are both set to 0.5. The point feature extraction backbone is a Minkowski 3D U-Net [7] pre-trained on large-scale 3D data. For our own test samples, 10,000 points are uniformly sampled from each mesh, covering both normal and anomalous cases for evaluation.

4.2. Evaluation results

We present quantitative results of object-level AUC-ROC performance across three 3D anomaly detection datasets. Specifically, results on Anomaly-ShapeNet are shown in Tab. 1 with anomaly portions from 1% to 7%, results on Real3D-AD in Tab. 2 with portions from 1% to 5.4%, and results on our Industry dataset in Tab. 3 where the anomaly ratio varies from below 1% to over 80%. Since each class in the proposed test set contains only one instance, object-level metrics are not applicable; thus, we use point-level metrics. For point-level results on the other two datasets, please refer to the supplementary material. These datasets collectively cover a broad range of anomaly densities and spatial distributions.

As shown in Tab. 1, our approach achieves state-of-the-art average performance across 40 classes. Although it is slightly lower (by only 3%–5%) than PO3AD [34] on a few cases such as "bottle0", "eraser0", and "helmet0", the overall results remain dominant. Note that class-specific weights are unavailable in PO3AD, preventing exact reproduction; thus, we report the paper's original results. As shown in Tab. 2, although a few classes (e.g., "Chicken", "Diamond", and "Fish") have the second-best method outperforming our method, our method still averages a 7.5% margin over the second-best method.

On our Industry dataset (Tab. 3) with challenging angular misalignment and planar displacements, our model substantially outperforms all baselines. R3D-AD [41] fails on most samples, while our model exceeds PO3AD [34] by over 50% in point-level AUC-ROC, demonstrating strong robustness under real industrial conditions. We further provide Point-level AUC-ROC(↑) results for Anomaly-ShapeNet and Real3DAD in the supplementary file.

Qualitative comparisons in Fig. 3 cover diverse anomaly types including angular shifts, bending, concavity, surface sinks, and bulges. While R3D-AD [41] produces unstable detections and PO3AD [34] yields many false positives, our method consistently delivers accurate localization aligned with ground truth across all datasets.

4.3. Ablation study

Time complexity & memory Our model achieves similar inference speed (178 ms) to PO3AD (172 ms) and R3DAD (183 ms) while offering stronger performance on global anomalies. In terms of GPU memory usage, R3DAD requires 2318 MB, PO3AD 2013 MB, and ours 2125 MB for the same point input. As shown in Tab. 4, the 3D UNet backbone dominates computation (51.7%), followed by the RoPE cross-attention decoder (29.4%). FPS patchification accounts for 14.4% of inference time, while the lightweight MLP head contributes minimal overhead (4.5%). All timings were measured on a single NVIDIA RTX 3090 GPU with a batch size of 1.

Hyperparameter analysis Fig. 4 presents a sensitivity analysis of multi-scale patch settings on Real3D-AD [17]. Increasing the **voxel size** consistently improves both object-level and point-level AUC-ROC, with performance saturating beyond 128^3, indicating sufficient geometric detail capture. The **patch number**, evaluated at voxel resolution 256^3 and patch size 16, yields steady gains before plateauing after 128 patches, suggesting diminishing returns from finer spatial partitioning. In contrast, the **patch size**, tested at voxel resolution 256^3 and patch number 32, peaks at size 64 and gradually declines at larger scales (128, 256), as oversized patches blur fine-grained anomaly boundaries. Overall, moderate voxel resolution ($\geq 128^3$), sufficient patch numbers (≥ 128), and a balanced patch size (64) provide the best trade-off between local detail and global representation for robust anomaly detection and localization.

Patchification strategy We evaluate the impact of different patchification strategies on performance, including semantic-part-based, FPS voxel-based, 3D grid, FPS sphere, and multi-scale sphere patchification. For semantic-part-based patchification, we use Find3D [22] with part-text prompts and PartField [18] on meshes, followed by point-cloud sampling with part labels. As shown, the multi-scale patchification achieves the best anomaly detection performance, while the semantic-part-based variant also attains comparable AUC-ROC to ISMP [16] in Tab. 2. However, large semantic parts lack sufficient geometric granularity for fine feature matching. Additional details on patchification strategies are provided in the supplementary materials.

Ablation of design choices. As shown in Tab. 6, removing negative augmentation significantly reduces the ability

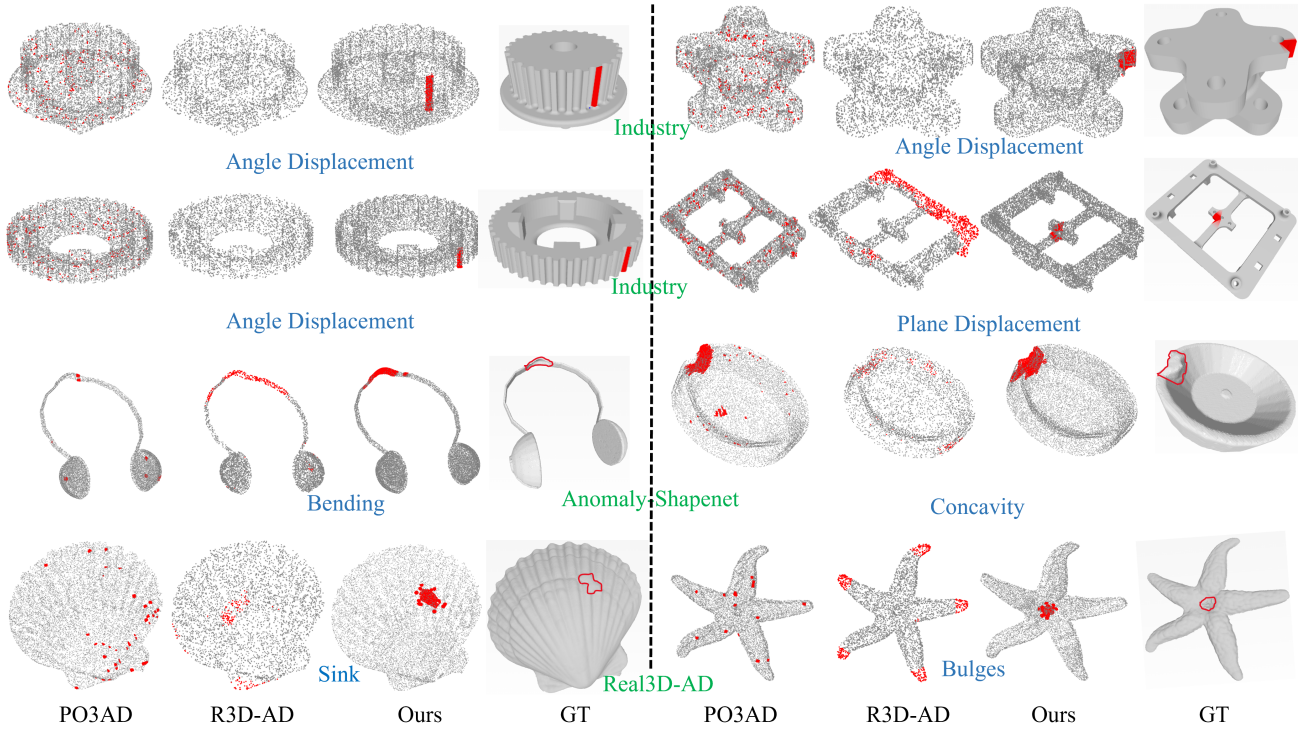


Figure 3. Qualitative comparison on three datasets (green letters): our crafted industrial components (1st–2nd rows), Real3D-AD (3rd), and Anomaly-ShapeNet (4th). The dashed line indicates different object classes. Representative anomaly types (in blue) include displacement, bending, concavity, sinks, and bulges. Anomaly points are highlighted in red. Ground truth uses red masks (first two rows) or red contours (last two rows) to indicate anomaly regions overlaid with an anomalous mesh. Each column shows results from a specific method.

Table 1. Comparison of object-level AUC-ROC results (%) of various methods on the Anomaly-ShapeNet dataset.

Method	ashtray0	bag0	bottle0	bottle1	bottle3	bowl0	bowl1	bowl2	bowl3	bowl4	bowl5	bucket0	bucket1	cap0
CPMF [5]	35.3	64.3	52.0	48.2	40.5	78.3	63.9	62.5	65.8	68.3	68.5	48.2	60.1	60.1
Reg3D-AD ([17])	59.7	70.6	48.6	69.5	52.5	67.1	52.5	49.0	34.8	66.3	59.3	61.0	75.2	69.3
IMRNet [14]	67.1	66.0	55.2	70.0	64.0	68.1	70.2	68.5	59.9	67.6	71.0	58.0	77.1	73.7
R3D-AD [41]	83.3	72.0	73.3	73.7	78.1	81.9	77.8	74.1	76.7	74.4	65.6	68.3	75.6	82.2
ISMP [16]	86.5	73.4	72.2	86.9	74.0	76.2	70.2	70.6	85.1	75.3	73.3	54.5	68.3	67.2
PO3AD [34]	100.0	83.3	**90.0**	93.3	**92.6**	92.2	82.9	83.3	**88.1**	**98.1**	84.9	85.3	78.7	87.7
Ours	**100.0**	**85.2**	87.9	**93.8**	91.3	**94.8**	**88.4**	**86.7**	86.9	97.1	**89.3**	**87.0**	**83.9**	**90.5**

Method	cap3	cap4	cap5	cup0	cup1	eraser0	headset0	headset1	helmet0	helmet1	helmet2	helmet3	jar0	micro.
CPMF [5]	55.1	55.3	69.7	49.7	49.9	68.9	64.3	45.8	55.5	58.9	46.2	52.0	61.0	50.9
Reg3D-AD [17]	72.5	64.3	46.7	51.0	53.8	34.3	53.7	61.0	60.0	38.1	61.4	36.7	59.2	41.4
IMRNet [14]	77.5	65.2	65.2	64.3	75.7	54.8	72.0	67.6	59.7	60.0	64.1	57.3	78.0	75.5
R3D-AD [41]	73.0	68.1	67.0	77.6	75.7	89.0	73.8	79.5	75.7	72.0	63.3	70.7	83.8	76.2
ISMP [16]	77.5	66.1	77.0	55.2	85.1	52.4	47.2	84.3	61.5	60.3	56.8	52.2	66.1	60.0
PO3AD [34]	85.9	79.2	67.0	87.1	83.3	**99.5**	80.8	92.3	76.2	**96.1**	86.9	75.4	86.6	77.6
Ours	**86.4**	**79.8**	**80.1**	**89.2**	**85.6**	94.0	**81.7**	**92.8**	**78.1**	93.7	**88.4**	**83.9**	**88.7**	**80.3**

Method	shelf0	tap0	tap1	vase0	vase1	vase2	vase3	vase4	vase5	vase7	vase8	vase9	Average
CPMF [5]	68.5	35.9	69.7	45.1	34.5	58.2	58.2	51.4	61.8	39.7	52.9	60.9	55.9
Reg3D-AD [17]	68.8	67.6	64.1	53.3	70.2	60.5	65.0	50.0	52.0	46.2	62.0	59.4	57.2
IMRNet [14]	60.3	67.6	69.6	53.3	75.7	61.4	70.0	52.4	67.6	63.5	63.0	59.4	66.1
R3D-AD [41]	69.6	73.6	90.0	78.8	72.9	75.2	74.2	63.0	75.7	77.1	72.1	71.8	74.9
ISMP [16]	70.1	84.4	67.8	68.7	53.4	77.3	62.2	54.6	58.0	74.7	73.6	82.3	69.1
PO3AD [34]	57.3	74.5	68.1	85.8	74.2	95.2	82.1	67.5	85.2	**96.6**	73.9	83.0	83.9
Ours	**74.1**	**88.2**	**92.4**	**89.5**	**82.9**	**96.3**	**84.7**	**80.4**	**89.0**	93.1	**82.5**	**87.2**	**87.6**

Table 2. Object-level AUC-ROC results (%) of methods on the Real3D-AD dataset.

Category	CPMF [5]	Reg3D-AD [17]	IMRNet [14]	R3D-AD [41]	Group3AD [42]	PO3AD [34]	ISMP [16]	Ours
Airplane	70.1	71.6	76.2	77.2	74.4	80.4	85.8	**87.5**
Car	55.1	69.7	71.1	69.3	72.8	65.4	73.1	**75.6**
Candy	55.2	68.5	75.5	71.3	84.7	78.5	85.2	**86.8**
Chicken	50.4	**85.2**	78.0	71.4	78.6	68.6	71.4	82.3
Diamond	52.3	90.0	90.5	68.5	93.2	80.1	**94.8**	90.1
Duck	58.2	58.4	51.7	90.9	67.9	82.0	71.2	**91.6**
Fish	55.8	91.5	88.0	69.2	**97.6**	85.9	94.5	92.4
Gemstone	58.9	41.7	67.4	66.5	53.9	69.3	46.8	**70.1**
Seahorse	72.9	76.2	60.4	72.0	84.1	75.6	72.9	**86.3**
Shell	65.3	58.3	66.5	84.0	58.5	80.0	62.3	**83.7**
Starfish	70.0	50.6	67.4	70.1	56.2	75.8	66.0	**78.5**
Toffees	39.0	82.7	77.4	70.3	79.6	77.1	84.2	**85.9**
Average	58.6	70.4	72.5	73.4	75.1	76.5	76.7	**84.2**

Table 3. Point-level AUC-ROC results (%) of methods on the Industry dataset.

Method	Bearing Cap	Dynamixel Plate	Franka Connector	Hull	M4 Tensioning Spool	Teeth Wrist Servo	Teeth Carpals	Wrist Pulley Base	Avg
R3D-AD [41]	—	—	—	19.1	20.8	—	—	—	20.0
PO3AD [34]	21.6	25.3	18.2	36.5	5.9	7.8	5.1	30.9	18.9
ISMP [16]	53.5	35.8	51.1	68.4	60.3	57.9	49.9	37.2	51.8
Ours	**67.8**	**64.3**	**75.6**	**78.1**	**82.3**	**85.9**	**89.5**	**61.2**	**75.6**

Table 4. Inference time breakdown of model components.

Module	Time (ms)	Percentage (%)
3D UNet (MinkUNet34C)	92.1	51.7
RoPE Cross-Attention Decoder	52.3	29.4
MLP-based Head	8.0	4.5
FPS Patchification	25.7	14.4
Total	**178.1**	**100.0**

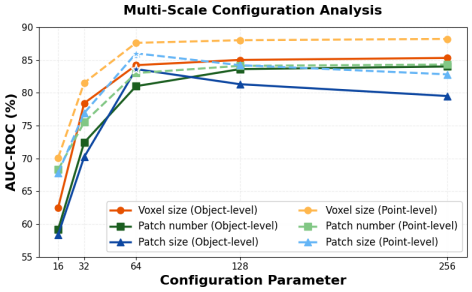
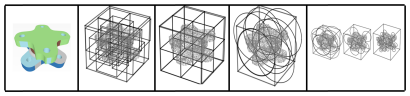


Figure 4. AUC-ROC performance under varying configuration parameters. Three parameter groups are compared: voxel size, patch number, and patch size. Each parameter is evaluated at object-level (solid lines) and point-level (dashed lines).

Table 5. Comparison of patch generation strategies (with figure example over column headers) in terms of object-level and point-level AUC-ROC performance (%) on Real3D-AD dataset.

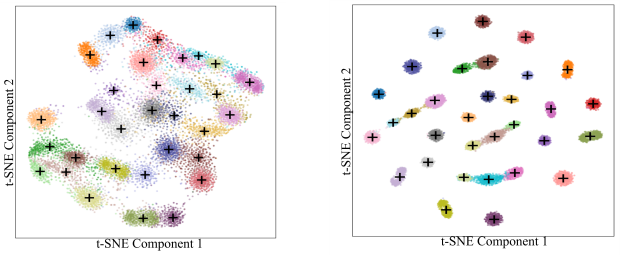


Metric	Semantic Parts	FPS Voxels	3D Grids	FPS Spheres	Multi-scale Spheres
Object (%)	75.1	81.4	80.3	81.9	**84.2**
Point (%)	78.2	82.5	81.7	82.6	**85.3**

of the model to bridge the gap between normal and anomalous samples, resulting in degraded performance. We then compare three patch feature extraction methods: (1) direct pooling of point features $\in \mathbb{R}^{32}$, and (2) computing the mean point coordinates $\in \mathbb{R}^3$ within each patch and querying the pre-trained encoder for its feature, and (3) averaging all patch point features $\in \mathbb{R}^{32}$. Among these, the second approach, using mean point coordinates as queries, achieves the best performance. Finally, RoPE-based cross-attention proves essential for effective patch–point feature fusion: removing ML modulation of normal and anomalous features reduces accuracy by about 4%, while disabling point-wise fusion leads to a significant drop of more than 10%. Empirically, selecting the most relevant scale by max in Eq. (2) outperforms fusing all scales (e.g., 84.2% vs. 81.2% AUC-ROC on Real3D-AD), since anomaly scales vary across categories and forced fusion can dilute discriminative features.

Table 6. Ablation study on the effectiveness of individual modules and pre-processing designs on the Real3D-AD [17] dataset (Best viewed when zoomed in.).

Metric	No Neg-Aug	Pooling	Mean Point	Mean Feature	RoPE Cross-Atten
AUC-ROC	68.2	77.8	79.4	76.5	84.2
AUC-PR	65.1	75.9	73.6	78.3	82.7
Metric	**Vanilla Cross-Atten**	**No MLP Modul**	**No Cross-Atten**	**Multi-scale Modulation**	**Full**
AUC-ROC	79.1	80.3	72.6	81.2	**84.2**
AUC-PR	77.4	78.5	70.8	79.3	**82.7**



(a) W/o Patch Feature Fusion (b) Full Model

Figure 5. t-SNE visualization of point feature distributions. '+' denotes patch centers; colors indicate 32 patch membership. (a) Without fusion, features scatter with poor separation, lacking discriminative structure despite close patch centers. (b) With patch feature modulation and fusion, clusters are well-separated.

To better illustrate the benefits of patch features, we present the results in Fig. 5 for 32 patches on the "starfish" class. For the tests, the model is trained for an equivalent of 1000 epochs. Without fusion (Fig. 5a), point features scatter with poor separation, indicating weak discriminative power for reliable anomaly localization. With our patch feature modulation and fusion, the features form well-separated clusters with clear margins (Fig. 5b), enabling more accurate anomaly discrimination for regression. For additional ablation results on patch parameters such as patch size and patch number, please refer to the supplementary material.

5. Conclusion

We propose a multi-scale patch-based framework for shape anomaly detection. We further introduce a real industrial test set with planar and angular displacement defects.

Limitations & future work. Our method remains sensitive to sampling randomness. Future work will explore incorporating topological or semantic part cues to produce more consistent patches and scale to real industrial data.

Acknowledgements

The first two authors acknowledge the financial support from the University of Melbourne through the Melbourne Research Scholarship. This research was supported by the University of Melbourne's Research Computing Services and the Petascale Campus initiative.

References

[1] Inigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8219–8228, 2021. 2

[2] Paul Bergmann and David Sattlegger. Anomaly detection in 3d point clouds using deep geometric descriptors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2613–2623, 2023. 2

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 1, 2

[4] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. *arXiv preprint arXiv:2112.09045*, 2021. 1, 2

[5] Yunkang Cao, Xiaohao Xu, and Weiming Shen. Complementary pseudo multimodal feature for point cloud anomaly detection. *Pattern Recognition*, 156:110761, 2024. 7, 8

[6] Yuqi Cheng, Yunkang Cao, Guoyang Xie, Zhichao Lu, and Weiming Shen. Towards zero-shot point cloud anomaly detection: A multi-view projection framework. *arXiv preprint arXiv:2409.13162*, 2024. 2

[7] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 6

[8] Yu-Min Chu, Liu Chieh, Ting-I Hsieh, Hwann-Tzong Chen, and Tyng-Luh Liu. Shape-guided dual-memory learning for 3d anomaly detection. *Proceedings of Machine Learning Research*, 2023. 1

[9] Alex Costanzino, Pierluigi Zama Ramirez, Giuseppe Lisanti, and Luigi Di Stefano. Multimodal industrial anomaly detection by crossmodal feature mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17234–17243, 2024. 2

[10] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den 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*, pages 1705–1714, 2019. 2

[11] Matthäus Heer, Janis Postels, Xiaoran Chen, Ender Konukoglu, and Shadi Albarqouni. The ood blind spot of unsupervised anomaly detection. In *Medical Imaging with Deep Learning*, pages 286–300. PMLR, 2021. 2

[12] Teng Hu, Jiangning Zhang, Ran Yi, Yuzhen Du, Xu Chen, Liang Liu, Yabiao Wang, and Chengjie Wang. Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8526–8534, 2024. 1, 2

[13] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9664–9674, 2021. 1

[14] Wenqiao Li, Xiaohao Xu, Yao Gu, Bozhong Zheng, Shenghua Gao, and Yingna 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)*, pages 22207–22216, 2024. 1, 2, 5, 7, 8

[15] Zizhao Li, Xueyang Kang, Joseph West, and Kourosh Khoshelham. Out-of-distribution detection in 3d applications: a review. *arXiv preprint arXiv:2507.00570*, 2025. 2

[16] Hanzhe Liang, Guoyang Xie, Chengbin Hou, Bingshu Wang, Can Gao, and Jinbao Wang. Look inside for more: Internal spatial modality perception for 3d anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(5):5146–5154, 2025. 6, 7, 8

[17] Jiaqi Liu, Guoyang Xie, Ruitao Chen, Xinpeng Li, Jinbao Wang, Yong Liu, Chengjie Wang, and Feng Zheng. Real3d-ad: A dataset of point cloud anomaly detection. *Advances in Neural Information Processing Systems*, 36:30402–30415, 2023. 1, 3, 5, 6, 7, 8

[18] Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9704–9715, 2025. 6

[19] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20402–20411, 2023. 1, 2

[20] Kaifang Long, Guoyang Xie, Lianbo Ma, Jiaqi Liu, and Zhichao Lu. Revisiting multimodal fusion for 3d anomaly detection from an architectural perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12273–12281, 2025. 2

[21] Haodong Lu, Dong Gong, Shuo Wang, Jason Xue, Lina Yao, and Kristen Moore. Learning with mixture of prototypes for out-of-distribution detection. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[22] Ziqi Ma, Yisong Yue, and Georgia Gkioxari. Find any part in 3d, 2024. 6

[23] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution

detection? In *The Eleventh International Conference on Learning Representations*, 2023. 1

[24] Sen Pei, Xin Zhang, Bin Fan, and Gaofeng Meng. Out-of-distribution detection with boundary aware learning. In *European Conference on Computer Vision*, pages 235–251. Springer, 2022. 1

[25] Jianjian Qin, Chunzhi Gu, Jun Yu, and Chao Zhang. Teacher–student network for 3d point cloud anomaly detection with few normal samples. *Expert Systems with Applications*, 228:120371, 2023. 1

[26] Yuchen Rao, Yinyu Nie, and Angela Dai. Patchcomplete: Learning multi-resolution patch priors for 3d shape completion on unseen categories. *Advances in Neural Information Processing Systems*, 35:34436–34450, 2022. 2

[27] Hannah M Schlüter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. Natural synthetic anomalies for self-supervised anomaly detection and localization. In *European Conference on Computer Vision*, pages 474–489. Springer, 2022. 2

[28] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 4

[29] Guanxiong Sun, Yang Hua, Guosheng Hu, and Neil Robertson. Mamba: Multi-level aggregation via memory bank for video object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2620–2627, 2021. 2

[30] Chengjie Wang, Haokun Zhu, Jinlong Peng, Yue Wang, Ran Yi, Yunsheng Wu, Lizhuang Ma, and Jiangning Zhang. M3dm-nr: Rgb-3d noisy-resistant industrial anomaly detection via multimodal denoising. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2

[31] Yue Wang, Jinlong Peng, Jiangning Zhang, Ran Yi, Yabiao Wang, and Chengjie Wang. Multimodal industrial anomaly detection via hybrid fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8032–8041, 2023. 2

[32] Zi Wang, Katsuya Hotta, Koichiro Kamide, Yawen Zou, Chao Zhang, and Jun Yu. 3dkeyad: High-resolution 3d point cloud anomaly detection via keypoint-guided point clustering. *arXiv preprint arXiv:2507.13110*, 2025. 2

[33] Jingkang Yang, Kaiyang Zhou, and Ziwei Liu. Full-spectrum out-of-distribution detection. *International Journal of Computer Vision*, 131(10):2607–2622, 2023. 2

[34] Jianan Ye, Weiguang Zhao, Xi Yang, Guangliang Cheng, and Kaizhu Huang. Po3ad: Predicting point offsets toward better 3d point cloud anomaly detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1353–1362, 2025. 2, 5, 6, 7, 8

[35] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draema discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8330–8339, 2021. 2

[36] Jinsong Zhang, Qiang Fu, Xu Chen, Lun Du, Zelin Li, Gang Wang, xiaoguang Liu, Shi Han, and Dongmei Zhang. Out-of-distribution detection based on in-distribution data patterns memorization with modern hopfield energy. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[37] Ximiao Zhang, Min Xu, and Xiuzhuang Zhou. Realnet: A feature selection network with realistic synthetic anomaly for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16699–16708, 2024. 2

[38] Haotian Zheng, Qizhou Wang, Zhen Fang, Xiaobo Xia, Feng Liu, Tongliang Liu, and Bo Han. Out-of-distribution detection learning with unreliable out-of-distribution sources. *Advances in neural information processing systems*, 36:72110–72123, 2023. 1, 2

[39] Junbao Zhou, Ziqi Pang, and Yu-Xiong Wang. Rmem: Restricted memory banks improve video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18602–18611, 2024. 2

[40] Qihang Zhou, Jiangtao Yan, Shibo He, Wenchao Meng, and Jiming Chen. Pointad: Comprehending 3d anomalies from points and pixels for zero-shot 3d anomaly detection. *Advances in Neural Information Processing Systems*, 37: 84866–84896, 2024. 2

[41] Zheyuan Zhou, Le Wang, Naiyu Fang, Zili Wang, Lemiao Qiu, and Shuyou Zhang. R3d-ad: Reconstruction via diffusion for 3d anomaly detection. In *European conference on computer vision*, pages 91–107. Springer, 2024. 2, 5, 6, 7, 8

[42] Hongze Zhu, Guoyang Xie, Chengbin Hou, Tao Dai, Can Gao, Jinbao Wang, and Linlin Shen. Towards high-resolution 3d anomaly detection via group-level feature contrastive learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 4680–4689, 2024. 8

[43] David Zimmerer, Peter M Full, Fabian Isensee, Paul Jäger, Tim Adler, Jens Petersen, Gregor Köhler, Tobias Ross, Annika Reinke, Antanas Kascenas, et al. Mood 2020: A public benchmark for out-of-distribution detection and localization on medical images. *IEEE Transactions on Medical Imaging*, 41(10):2728–2738, 2022. 2

[44] Zuo Zuo, Jiahao Dong, Yao Wu, Yanyun Qu, and Zongze Wu. Clip3d-ad: Extending clip for 3d few-shot anomaly detection with multi-view images generation. *arXiv preprint arXiv:2406.18941*, 2024. 2