Nature3D-AD: Geometry-Aware Feature Learning for Natural-Growth 3D Anomaly Detection

Jian Ning[1] **Qin Zou**[1,2*] **Linchun Wu**[1] **Kunmo Li**[3] **Chi Chen**[2,4] **Zhen Dong**[4]
Bisheng Yang[4] **Qingquan Li**[4,5]

[1]School of Computer Science, Wuhan University, Wuhan, China
[2]School of Robotics, Wuhan University, Wuhan, China
[3]School of Computer Science and Technology, Dalian University of Technology, Dalian, China
[4]State Key Laboratory LIESMARS, Wuhan University, Wuhan, China
[5]Guangdong Key Laboratory of Urban Informatics, Shenzhen University, Shenzhen, China
{ningjian, qzou}@whu.edu.cn



Figure 1: **An overview of the NaturalGrowth dataset.** It is the first dataset for agricultural point cloud anomaly detection, with all samples acquired through real-world scanning. We showcase the intricate details of the collected samples, which accurately reflect the immense diversity and natural growth variations of agricultural products.

Abstract

3D anomaly detection is a crucial process in agricultural production chains, playing a vital role in automated processing and ensuring food safety. However, existing 3D anomaly detection datasets are primarily designed for industrial scenarios with standardized geometric shapes. In contrast, agricultural products originate from natural growth, exhibiting significant variations in shape, size, and texture. Consequently, the definition of "normal" is far less explicitly constrained than that of industrial components, rendering current industrial datasets and detection methods inadequate for this application. To bridge this gap, we make contributions from both data and method perspectives. On the data side, we introduce NaturalGrowth, the first dataset dedicated to agricultural point cloud anomaly detection. On the method side, we propose Nature3D-AD, which incorporates two novel modules. Anomaly-Sensitive Positional Encoding (ASPE) integrates local curvature and density information to provide geometry-aware positional representations. Geometry-Aware Attention (GAA) injects geometric biases in the spatial domain and captures global structural patterns through a spectral FFT branch. Extensive

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

experiments on NaturalGrowth, Anomaly-ShapeNet, and Real3D-AD demonstrate that Nature3D-AD achieves state-of-the-art performance. The dataset and code will be released at https://ningxiaojian.github.io/Nature3D-AD/.

1 Introduction

Agricultural product quality inspection [15, 39] is a critical step in agricultural production chains. Current mainstream approaches rely on 2D images [20, 30, 40], but they are susceptible to lighting and viewpoint variations, and cannot perceive depth information, limiting their ability to detect geometric defects such as dents and cracks. In contrast, 3D point clouds directly represent the complete geometric structure of objects, making them more suitable for detecting surface deformations and structural anomalies. In recent years, 3D point cloud anomaly detection has demonstrated great potential in industrial quality control [6, 31, 38], with multiple datasets [21, 1, 14] and methods [9, 32, 37] advancing the field rapidly. However, applying this technology to agricultural quality inspection still faces challenges from both data and method perspectives.

On the data side, existing 3D point cloud anomaly detection datasets [1, 14], such as Real3D-AD [21] and Anomaly-ShapeNet [13], are almost exclusively designed for industrial scenarios, where the objects of interest are manufactured parts with standardized geometric shapes. Unlike industrial products, agricultural products originate from natural growth and have no standardized geometric template. Even normal samples within the same category exhibit substantial differences in shape, size, and surface texture. Furthermore, the anomaly morphologies of agricultural products are extremely diverse, ranging from tiny holes to large-scale rot, with a vast span of scales, and many anomalies are geometrically similar to natural growth variations. These characteristics render existing industrial datasets and detection methods [5, 27, 12] inadequate for agricultural scenarios, yet the field still lacks a dedicated 3D point cloud anomaly detection dataset, which severely hinders systematic research and method evaluation for 3D anomaly detection in agricultural settings.

On the method side, existing 3D point cloud anomaly detection methods generally follow either an embedding-based [9, 28, 2] or a reconstruction-based paradigm [37, 32, 3, 18], but face two key issues when applied to agricultural point clouds. First, the positional encodings in existing methods treat all regions uniformly and cannot perceive local geometric differences [33, 29, 10], causing the network to allocate equal attention to geometrically complex regions and flat regions during reconstruction, which results in insufficient reconstruction precision and inaccurate anomaly localization. Moreover, during the reconstruction process, agricultural products have no standard samples, so models must rely on self-referencing for reconstruction. However, existing attention mechanisms [4, 22] compute weights solely based on feature similarity, ignoring the spatial distance and geometric relationships between points, making them prone to erroneously matching features across unrelated regions, which leads to reconstruction distortion. This issue constrains the detection performance of existing methods on agricultural product point clouds.

To address the above issues, we make contributions from both data and method perspectives. On the data side, we introduce NaturalGrowth, the first dataset dedicated to agricultural 3D point cloud anomaly detection. As shown in Figure 1, it covers 31 diverse agricultural categories with 1,477 real-world scanned samples and 14 fine-grained anomaly types, accurately reflecting the rich variations arising from natural growth. On the method side, we propose Nature3D-AD, a detection framework incorporating Anomaly-Sensitive Positional Encoding (ASPE) and Geometry-Aware Attention (GAA) to address the two issues respectively. ASPE leverages local curvature and density to generate differentiated positional encodings for regions of varying geometric complexity through a learnable MLP, enabling the network to adaptively focus on structurally complex regions. GAA enhances self-attention from two complementary perspectives: the spatial branch injects geometric biases based on spatial distance and normal similarity, while the spectral branch captures global structural patterns through FFT, with both aggregated via gated fusion.

Our contributions are summarized as follows:

- We introduce NaturalGrowth, the first dataset for agricultural 3D point cloud anomaly detection, comprising 31 categories, 1,477 real-world scanned samples, and 14 fine-grained anomaly types with point-level annotations.

- We propose Nature3D-AD, a reconstruction-based framework that incorporates two novel modules. ASPE enables positional encodings to perceive local geometric differences, and GAA enhances attention with geometric awareness from both spatial and spectral perspectives.
- Extensive experiments on NaturalGrowth, Anomaly-ShapeNet, and Real3D-AD demonstrate that Nature3D-AD achieves state-of-the-art performance across all datasets.

2 Related Work

2.1 3D Anomaly Detection Datasets

The development of 3D anomaly detection datasets has driven rapid progress in this field in recent years. MVTec 3D-AD [1] is one of the earliest 3D anomaly detection datasets, providing multi-modal data with RGB images and depth maps. Subsequently, Real3D-AD [21] collected 1,254 real industrial part point cloud samples across 12 categories using a high-precision scanner, becoming an important dataset for pure point cloud anomaly detection. Anomaly-ShapeNet [13] generated anomalies procedurally based on ShapeNet models, providing a large-scale synthetic dataset. MulSen-AD [14] further introduced multi-sensor fusion data combining RGB, point cloud, and infrared thermography. Open-Industry [19] collects real defects from production lines for open-set supervised 3D anomaly detection. However, all the above datasets are designed for industrial manufacturing scenarios, where the objects of interest are manufactured parts with standardized geometric shapes. Although MVTec 3D-AD includes a few agricultural categories such as carrot, peach, and potato, they are captured from a single fixed viewpoint that covers only the sensor-facing surface, with anomalies annotated at the pixel level, while MANTA [7] covers agricultural grains such as wheat and maize only in the form of 2D multi-view images. Due to their natural growth characteristics, agricultural products fundamentally differ from industrial products in terms of geometric diversity, anomaly morphology, and intra-class variation, and there is currently no 3D point cloud anomaly detection dataset dedicated to agricultural scenarios.

2.2 3D Anomaly Detection Methods

Existing 3D point cloud anomaly detection methods [6, 16, 26] generally follow either an embedding-based or a reconstruction-based paradigm. Embedding-based methods typically leverage pre-trained 3D backbone networks to extract features and detect anomalies by constructing a feature memory bank of normal samples. For example, BTF [9] and CPMF [2] utilize pre-trained features and nearest-neighbor retrieval for anomaly scoring. Reg2Inv [34] and RIF [17] learn rotation-invariant features to enhance feature discriminability under pose variations. Reconstruction-based methods learn a generative model from normal data and localize anomalies through reconstruction residuals. PO3AD [32] proposes a point-offset prediction-based reconstruction method and introduces a normal-vector-guided pseudo-anomaly augmentation strategy. PA3AD [24] further generates physically plausible pseudo-defects such as holes, cracks, and bends to better approximate real anomalies. Learning from synthesized anomalies has likewise proven effective in 2D anomaly detection [35, 8], and our method adopts the pseudo-anomaly generator of PO3AD rather than proposing a new one. PASDF [36] leverages a continuous signed distance field representation to achieve pose-invariant anomaly detection and repair. Although these methods have achieved promising performance on industrial datasets, they face challenges when processing agricultural point clouds, as the high geometric complexity and the absence of standard templates make it difficult for existing methods to effectively distinguish normal variations from genuine anomalies.

3 NaturalGrowth Dataset

As illustrated in Figure 2, our data acquisition pipeline consists of three stages. We employ an industrial-grade structured-light 3D scanning system integrating a texture camera, a structured-light camera, and a motorized turntable to simultaneously capture high-precision geometry and surface texture. Each sample is placed on the turntable and scanned from diverse viewpoints via multi-angle rotation and flipping to achieve comprehensive surface coverage. The partial scans are then automatically aligned and fused into a high-fidelity complete point cloud.

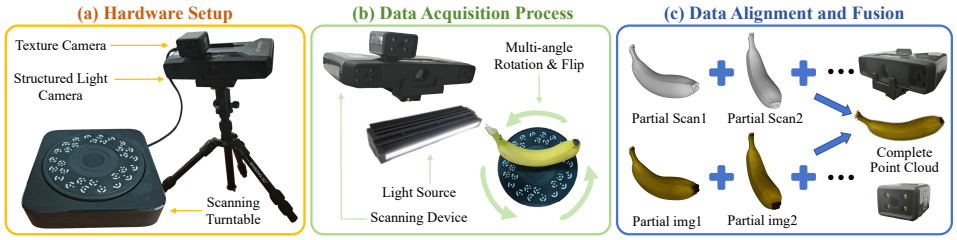


Figure 2: **Data acquisition and alignment pipeline for the NaturalGrowth dataset.** We utilize a 3D scanning system (a) to capture partial geometries and textures via multi-angle rotation (b), which are then aligned and fused into high-fidelity complete point clouds (c).

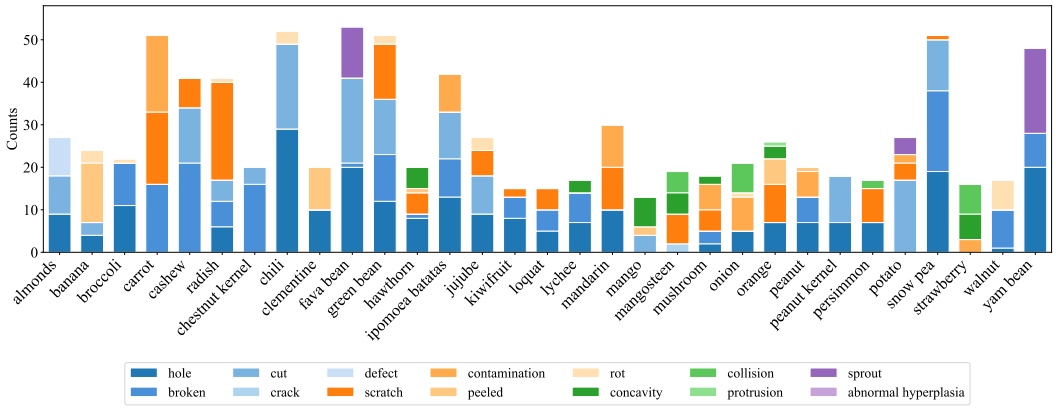


Figure 3: **Distribution of anomaly types in NaturalGrowth.** The stacked bar chart details the number of anomalous point clouds across 31 agricultural categories. Different colors denote the 14 fine-grained anomaly types, demonstrating the rich diversity and complex distribution of natural defects.

3.1 Dataset Composition and Annotation

NaturalGrowth comprises 1,477 real-world scanned samples spanning 31 common agricultural categories, covering a wide spectrum of morphologies from smooth spherical shapes (*e.g.*, orange, onion), elongated structures (*e.g.*, carrot, green bean), to highly irregular surfaces (*e.g.*, broccoli, walnut). For each category, we collect both normal and defective samples. The defects encompass naturally occurring anomalies (*e.g.*, wormholes, rot) as well as physical damage from harvesting and transportation (*e.g.*, bruising, breakage). The training set of each category contains 4 normal samples, while the test set consists of a mixture of normal and anomalous samples. After multi-view scanning and registration, the resulting point clouds range from approximately 10,000 to 472,000 points per sample, with an average of roughly 155,000 points.

The dataset covers 14 fine-grained anomaly types, including hole, cut, defect, contamination, rot, collision, sprout, broken, crack, scratch, peeled, concavity, protrusion, and abnormal hyperplasia, whose distribution is detailed in Figure 3. All anomalous samples are annotated at the point level using CloudCompare with binary labels (normal/anomalous), and each sample undergoes at least one round of cross-review.

3.2 Dataset Statistics and Characteristics

Compared with existing industrial datasets, NaturalGrowth presents two distinctive characteristics. First, it exhibits extreme intra-class variation, as the natural growth of agricultural products leads to substantial differences in shape and size even among normal samples of the same category, with no standardized geometric template available. Second, the anomaly scale span is remarkably large. The average proportion of anomalous points per object is 3.86% (median 2.52%, standard deviation

4.36%), yet the range varies from as low as 0.11% to as high as 35.15%, demanding algorithms to simultaneously possess fine-grained local perception and macro-level global awareness.

4 Methodology

4.1 Pipeline Overview

As illustrated in Figure 4, Nature3D-AD adopts a reconstruction-based anomaly detection paradigm. Given an input point cloud $P = \{p_i\}_{i=1}^{N} \in \mathbb{R}^{N\times 3}$, pseudo-anomalies are first generated via PA-Gen [32], and feature tokens are extracted by a frozen PointMAE [25] backbone. These tokens are enhanced with ASPE and fed into a Transformer reconstruction network composed of GAA and FFN layers. The model is optimized by minimizing the MSE between original and reconstructed features, and anomaly scores are computed from their per-point difference during inference.

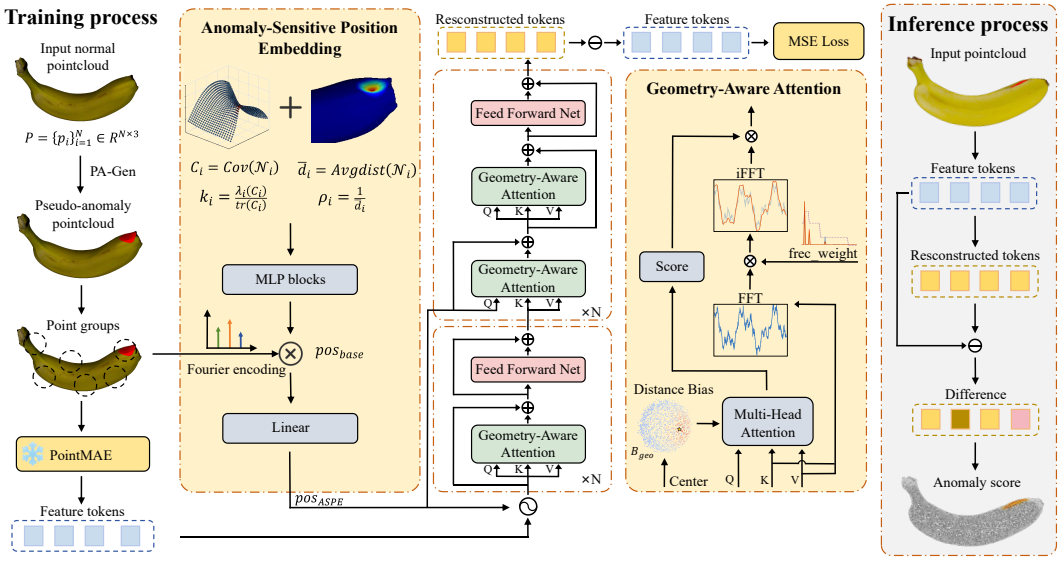


Figure 4: **The overview of Nature3D-AD method.** During training, pseudo-anomalous point clouds are generated from the input via PA-Gen, and feature tokens are extracted using PointMAE. The tokens are then fed into a reconstruction network equipped with ASPE and GAA. During inference, anomalies are detected by comparing the original features with the reconstructed ones.

4.2 Anomaly-Sensitive Positional Encoding (ASPE)

In the domain of agricultural point cloud anomaly detection, test samples exhibit highly complex morphologies, typically featuring inherently high-curvature regions. This geometric diversity makes it extremely difficult for networks to distinguish between natural geometric variations and anomalous structural damage. To address this issue, we propose Anomaly-Sensitive Positional Encoding (ASPE), which adaptively modulates the intensity of positional encodings by explicitly exploiting local geometric properties, thereby providing critical spatial cues to the network.

Specifically, for the i-th group center $c_i \in \mathcal{C}$, we construct its local neighborhood $\mathcal{N}_i$ using k-nearest neighbors. The local curvature $\kappa_i = \lambda_0 / \sum_{j=0}^{2} \lambda_j$ is estimated via PCA on the covariance matrix of $\mathcal{N}_i$, where λ_0 is the smallest eigenvalue. The local point density ρ_i is first normalized to $[0, 1]$ and then converted to a density deviation measure via $2|\rho_i - 0.5|$, so that both overly sparse and overly dense regions receive high responses.

These geometric properties are concatenated and fed into a learnable MLP followed by a Sigmoid function to produce a weight vector $W_i \in \mathbb{R}^D$. Crucially, ASPE does not directly amplify positional encodings based on raw curvature or density values. Instead, the network learns through training which combinations of geometric properties warrant stronger positional representations. The final

positional encoding P_i^{ASPE} is obtained by modulating the base Fourier encoding P_i^{base}:

$$P_i^{\text{ASPE}} = P_i^{\text{base}} \odot (\gamma + \alpha \cdot W_i), \tag{1}$$

where $\odot$ denotes element-wise multiplication, γ is a base offset, and α is a learnable scaling factor. Since W_i is bounded in $[0, 1]$ by the Sigmoid function, the modulation weight is constrained to $[\gamma, \gamma + \alpha]$, ensuring that positional information is never entirely suppressed for any region.

4.3 Geometry-Aware Attention (GAA)

To enable the attention mechanism to perceive spatial distance and geometric relationships between points, we design the GAA module, which enhances feature learning from both spatial and spectral perspectives and adaptively aggregates them via a gated fusion mechanism.

Spatial Branch with Geometric Biases. We constrain feature matching by injecting explicit geometric biases into the self-attention matrix. For any two group centers c_i and c_j with their estimated normal vectors n_i and n_j (computed via PCA on local neighborhoods), we construct a distance bias to prioritize neighboring elements and a normal similarity bias to favor smooth surface transitions. The normalized Euclidean distance $\bar{d}_{i,j} = \|c_i - c_j\|_2 / (\max_{i,j} \|c_i - c_j\|_2 + \epsilon)$ is mapped to an exponentially decaying prior, and the cosine similarity $\langle n_i, n_j \rangle$ reflects angular alignment, where ϵ is a small constant for numerical stability. Both are projected via MLPs (Φ_{dist} and Φ_{norm}) to form the combined geometric bias matrix:

$$\mathbf{B}_{geo} = w_d \Phi_{dist}(\exp(-\tau \cdot \bar{d}_{i,j})) + w_n \Phi_{norm}(\langle n_i, n_j \rangle), \tag{2}$$

where w_d, w_n, and τ are learnable scaling parameters. The geometric bias is then injected into the standard self-attention to obtain the spatially aggregated feature H^{spa}:

$$H^{spa} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_h}} + \mathbf{B}_{geo}\right) V W_O, \tag{3}$$

where Q, K, V are the query, key, and value matrices obtained via linear projections from the input features, d_h is the dimension per attention head, and W_O is the output projection matrix.

Spectral Branch. To ensure the locally learned patterns remain coherent within the object's overall morphological context, we introduce an FFT branch. This branch applies a FFT with orthogonal normalization along the token sequence dimension to the value features $\mathbf{V} \in \mathbb{R}^{B \times G \times C}$, transforming them from the spatial domain to the frequency domain. In the frequency domain, each frequency component encodes global relationships across all groups, thus providing a global receptive field. A learnable per-channel frequency weight $\omega \in \mathbb{R}^C$ then re-weights the spectrum, enabling the network to adaptively enhance or suppress frequency responses across different channels. The re-weighted spectrum is transformed back to the spatial domain via inverse FFT and passed through a two-layer MLP for non-linear projection, yielding the spectral feature H^{freq}, where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the real FFT and inverse real FFT operations:

$$H^{freq} = \Phi_{freq}\left(\mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{V}) \odot \omega\right)\right). \tag{4}$$

Gated Fusion. To adaptively balance local geometric precision and global structural integrity, the features from both domains are aggregated via a learnable gating parameter β:

$$H^{out} = \sigma(\beta) \cdot H^{spa} + (1 - \sigma(\beta)) \cdot H^{freq}. \tag{5}$$

where $\sigma(\cdot)$ is the Sigmoid function.

4.4 Training Objective and Anomaly Scoring

During training, the frozen PointMAE extracts the target features $F \in \mathbb{R}^{G \times D}$ from the uncorrupted normal point cloud. The reconstruction network instead takes the features of the pseudo-anomalous point cloud as input and reconstructs them into $\hat{F} \in \mathbb{R}^{G \times D}$, so that the anomalous input is restored to the uncorrupted representation F. The model is optimized by minimizing the MSE loss:

$$\mathcal{L}_{MSE} = \frac{1}{G} \sum_{i=1}^{G} \|\hat{F}_i - F_i\|_2^2. \tag{6}$$

During inference, the anomaly score for each group center is defined as the L2 distance between its original and reconstructed features. To obtain point-level localization, group-level scores are propagated to individual points via KNN interpolation with inverse distance weighting. The object-level anomaly score is obtained by applying average pooling to the point-level scores followed by a max operation.

Table 1: O-AUROC performance of different methods on NaturalGrowth across 31 categories, where best and second-place results are highlighted in **red** and **blue**, respectively.

O-AUROC											
Method	almonds	banana	broccoli	carrot	cashew	radish	chestnut	chili	clementine	fava bean	green bean
BTF(Raw) (CVPR23)	0.359	0.273	0.327	0.324	0.337	0.393	0.708	0.414	0.515	0.894	0.458
BTF(FPFH) (CVPR23)	0.424	0.503	0.409	0.424	0.610	0.488	0.542	0.419	0.880	0.544	0.459
M3DM (CVPR23)	0.329	0.414	0.476	0.461	0.372	0.399	0.700	0.397	0.610	0.695	0.545
PatchCore(FPFH) (CVPR22)	0.831	0.409	0.439	0.421	0.840	0.453	0.819	0.434	0.830	0.942	0.585
PatchCore(PointMAE) (CVPR22)	0.770	0.620	0.512	0.526	0.735	0.359	0.642	0.689	0.670	0.974	0.421
CPMF (PR24)	0.685	0.318	0.724	0.451	0.770	0.354	0.542	0.505	0.780	0.968	0.517
Reg3D-AD (NeurIPS23)	0.657	0.570	0.303	0.565	0.619	0.482	0.612	0.581	0.755	0.965	0.554
R3D-AD (ECCV24)	0.384	0.359	0.921	0.276	0.441	0.525	0.596	0.570	1.000	0.579	0.630
PO3AD (CVPR25)	0.845	0.479	0.482	0.343	0.569	0.521	0.588	0.630	0.865	0.669	0.574
MC3D-AD (IJCAI25)	0.887	0.524	0.715	0.642	0.732	0.833	0.696	0.747	1.000	0.895	0.688
ISMP (AAAI25)	0.426	0.540	0.455	0.577	0.774	0.613	0.566	0.588	0.767	0.744	0.610
Reg2Inv (NeurIPS25)	0.778	0.479	0.485	0.568	0.544	0.550	0.658	0.500	0.945	0.823	0.467
PASDF (ICCV25)	0.589	0.544	0.896	0.603	0.648	0.606	0.627	0.623	0.630	0.597	0.537
Mulsen-AD (CVPR25)	0.269	0.461	0.521	0.442	0.309	0.324	0.727	0.426	0.680	0.704	0.531
Simple3D (AAAI26)	0.722	0.393	0.636	0.511	0.736	0.458	0.908	0.425	0.965	0.979	0.716
Ours	0.725	0.458	0.670	0.841	0.910	0.765	1.000	0.752	1.000	0.972	0.845
Method	hawthorn	ipomoea	jujube	kiwifruit	loquat	lychee	mandarin	mango	mangosteen	mushroom	onion
BTF(Raw) (CVPR23)	0.550	0.688	0.642	0.538	0.653	0.647	0.736	0.636	0.658	0.596	0.610
BTF(FPFH) (CVPR23)	0.463	0.449	0.739	0.538	0.458	0.364	0.636	0.769	0.298	0.481	0.364
M3DM (CVPR23)	0.540	0.817	0.617	0.449	0.631	0.570	0.731	0.713	0.439	0.552	0.455
PatchCore(FPFH) (CVPR22)	0.440	0.396	0.831	0.351	0.567	0.662	0.794	0.538	0.509	0.494	0.658
PatchCore(PointMAE) (CVPR22)	0.450	0.604	0.636	0.631	0.556	0.732	0.656	0.580	0.596	0.694	0.481
CPMF (PR24)	0.500	0.714	0.568	0.347	0.636	0.382	0.561	0.762	0.368	0.444	0.489
Reg3D-AD (NeurIPS23)	0.400	0.610	0.593	0.409	0.476	0.559	0.664	0.706	0.632	0.614	0.528
R3D-AD (ECCV24)	0.067	0.376	0.584	0.538	0.720	0.504	0.964	0.622	0.123	0.627	0.476
PO3AD (CVPR25)	0.390	0.698	0.743	0.836	0.764	0.743	0.494	0.909	0.711	0.664	0.606
MC3D-AD (IJCAI25)	0.850	0.781	0.979	0.884	0.747	0.886	0.778	0.839	1.000	0.833	0.645
ISMP (AAAI25)	0.487	0.621	0.537	0.550	0.328	0.637	0.786	0.726	0.492	0.467	0.573
Reg2Inv (NeurIPS25)	0.883	0.405	0.674	0.587	0.724	0.838	0.850	0.839	0.798	0.963	0.623
PASDF (ICCV25)	0.615	0.623	0.634	0.604	0.673	0.634	0.707	0.630	0.606	0.550	0.554
Mulsen-AD (CVPR25)	0.587	0.815	0.582	0.516	0.471	0.680	0.658	0.762	0.693	0.688	0.567
Simple3D (AAAI26)	0.577	0.598	0.940	0.640	0.773	0.599	0.861	0.692	0.509	0.392	0.610
Ours	0.767	0.921	0.979	0.898	0.742	0.886	0.875	0.881	1.000	0.960	0.753
Method	orange	peanut	peanut kernel	persimmon	potato	snow pea	strawberry	walnut	yam bean	Average	Mean Rank
BTF(Raw) (CVPR23)	0.558	0.859	0.556	0.506	0.503	0.605	0.729	0.621	0.412	0.558	10.419
BTF(FPFH) (CVPR23)	0.630	0.666	0.344	0.588	0.897	0.479	0.492	0.636	0.728	0.539	11.113
M3DM (CVPR23)	0.579	0.772	0.715	0.510	0.491	0.570	0.750	0.618	0.627	0.566	10.274
PatchCore(FPFH) (CVPR22)	0.594	0.681	0.656	0.525	0.790	0.415	0.500	0.768	0.889	0.615	8.855
PatchCore(PointMAE) (CVPR22)	0.625	0.591	0.726	0.482	0.540	0.469	0.650	0.676	0.566	0.608	9.177
CPMF (PR24)	0.551	0.734	0.729	0.729	0.626	0.657	0.670	0.629	0.856	0.599	9.597
Reg3D-AD (NeurIPS23)	0.608	0.541	0.618	0.486	0.534	0.395	0.692	0.743	0.465	0.579	10.194
R3D-AD (ECCV24)	0.576	0.500	0.232	0.498	0.457	0.174	0.333	0.607	0.679	0.514	11.516
PO3AD (CVPR25)	0.603	0.338	0.538	0.600	0.948	0.528	0.813	0.662	0.751	0.642	7.484
MC3D-AD (IJCAI25)	0.748	0.759	0.817	0.541	0.927	0.651	0.827	0.801	0.897	0.792	3.387
ISMP (AAAI25)	0.575	0.522	0.524	0.524	0.629	0.519	0.639	0.615	0.579	0.580	9.871
Reg2Inv (NeurIPS25)	0.829	0.641	0.726	0.710	0.739	0.503	0.700	0.563	0.595	0.677	6.823
PASDF (ICCV25)	0.673	0.581	0.642	0.675	0.733	0.241	0.641	0.620	0.589	0.617	8.645
Mulsen-AD (CVPR25)	0.531	0.734	0.556	0.600	0.523	0.523	0.696	0.603	0.601	0.574	10.000
Simple3D (AAAI26)	0.885	0.772	0.740	0.678	0.928	0.583	0.308	0.820	0.907	0.686	6.210
Ours	1.000	0.756	0.958	0.957	0.965	0.526	0.741	0.938	0.993	0.853	2.435

Table 2: O-AUROC performance of different methods on Real3D 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	Mean Rank
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	11.875
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	12.000
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	14.458
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	12.583
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	13.167
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	13.250
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	9.458
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	8.833
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	8.250
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	7.542
PO3AD (CVPR25)	0.804	0.654	0.785	0.686	0.801	0.820	0.859	0.693	0.756	0.800	0.758	0.771	0.766	7.500
MC3D-AD (IJCAI25)	0.850	0.749	0.830	0.715	0.955	0.831	0.865	0.560	0.716	0.803	0.766	0.738	0.781	6.750
Reg2Inv (NeurIPS25)	0.818	0.758	1.000	0.944	1.000	0.750	0.672	0.735	0.532	0.692	0.841	0.626	0.781	6.292
PASDF (ICCV25)	0.628	0.959	0.788	0.739	0.894	0.658	0.989	0.634	1.000	0.850	0.617	0.866	0.802	6.500
Simple3D (AAAI26)	0.765	0.981	0.651	0.826	1.000	0.778	0.912	0.704	0.930	0.514	0.696	0.888	0.804	5.875
SeDiR (CVPR26)	0.860	0.783	0.819	0.729	0.948	0.862	0.938	0.627	0.674	0.779	0.854	0.845	0.810	5.542
Ours	0.950	0.947	0.956	0.809	0.965	0.951	0.742	0.766	0.935	0.834	0.963	0.827	0.887	3.125

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate Nature3D-AD on three datasets: (1) **NaturalGrowth** is the first 3D anomaly detection dataset for agricultural products, comprising 1,477 real-scanned samples across 31 categories with high intra-class diversity. (2) **Anomaly-ShapeNet** [13] is a synthetic dataset with

Table 3: O-AUROC performance of different methods on AnomalyShapeNet 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
ParchCore(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
ParchCore(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
PO3AD (CVPR25')	**1.000**	0.833	0.900	0.933	0.926	0.922	0.829	0.833	0.881	**0.981**	0.849	0.853	0.787	**0.877**
MC3D-AD (IJCAI25')	0.962	0.805	0.795	0.709	0.756	0.930	**0.978**	0.719	0.885	0.911	0.754	0.898	0.784	0.793
Reg2Inv (NeurIPS25')	0.900	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.807	0.656	0.585	0.852	0.818	0.813	0.902	0.659
Simple3D (AAAI26')	**0.995**	0.881	**0.976**	0.951	**1.000**	**1.000**	0.830	0.711	0.911	0.730	0.863	**0.959**	0.790	0.852
SeDiR (CVPR26')	0.976	0.886	0.929	0.916	**0.984**	0.959	0.956	**0.926**	**0.967**	**0.985**	**0.979**	0.911	**0.933**	**0.985**
Ours	0.966	**0.943**	0.971	**0.957**	0.914	**0.989**	**0.993**	**1.000**	**0.977**	0.929	**0.971**	**0.965**	**0.930**	**0.985**

Method	**cap3**	**cap4**	**cap5**	**cup0**	**cup1**	**eraser0**	**headset0**	**headset1**	**helmet0**	**helmet1**	**helmet2**	**helmet3**	**jar0**	**micro.0**
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
ParchCore(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
ParchCore(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
PO3AD (CVPR25')	0.859	0.792	0.670	0.871	0.833	**0.995**	0.808	**0.923**	0.762	0.961	0.869	0.754	0.866	0.776
MC3D-AD (IJCAI25')	0.701	0.835	0.761	0.743	0.952	0.776	0.862	0.886	0.672	**1.000**	0.609	**0.979**	0.971	**0.919**
Reg2Inv (NeurIPS25')	0.863	0.681	**0.902**	0.733	0.933	**1.000**	**1.000**	0.843	0.817	**0.986**	**0.875**	0.876	**1.000**	**1.000**
Simple3D (AAAI26')	0.867	0.912	0.804	**1.000**	0.824	**1.000**	0.982	**0.957**	0.690	0.719	0.754	0.655	0.905	**1.000**
SeDiR (CVPR26')	**0.989**	**0.996**	0.874	**0.995**	**1.000**	0.667	0.849	0.890	**0.965**	**1.000**	**0.870**	**1.000**	**1.000**	0.905
Ours	**0.968**	**0.961**	**0.957**	**1.000**	**0.985**	0.980	**0.991**	0.838	**0.852**	**1.000**	0.811	0.972	**0.990**	**1.000**

Method	**shelf0**	**tap0**	**tap1**	**vase0**	**vase1**	**vase2**	**vase3**	**vase4**	**vase5**	**vase7**	**vase8**	**vase9**	**Average**	**Mean Rank**
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	12.700
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	11.963
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	11.700
ParchCore(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	11.262
ParchCore(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	11.363
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	11.262
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	11.350
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	0.659	8.800
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	6.612
PO3AD (CVPR25')	0.573	0.745	0.681	0.858	0.742	0.952	0.821	0.675	0.852	**0.966**	0.739	0.830	0.839	4.838
MC3D-AD (IJCAI25')	0.841	**0.945**	**0.970**	0.821	0.857	0.929	0.761	0.876	0.976	0.938	0.670	0.736	0.842	4.875
Reg2Inv (NeurIPS25')	0.577	**0.948**	0.804	**0.996**	0.605	**1.000**	**0.845**	0.818	**1.000**	0.643	0.818	0.873	0.861	4.200
Simple3D (AAAI26')	0.748	0.703	0.596	0.954	0.824	0.871	0.812	0.864	0.962	0.895	0.855	0.815	0.860	4.138
SeDiR (CVPR26')	**0.887**	0.927	0.893	0.888	**0.971**	**0.976**	0.788	**0.924**	0.948	**1.000**	**0.873**	**0.942**	**0.933**	**2.737**
Ours	**0.954**	**0.948**	**0.996**	**0.991**	**0.871**	0.942	**0.909**	**0.909**	**0.991**	**1.000**	**0.972**	**0.990**	**0.957**	**2.200**

1,600 samples across 40 categories, constructed by injecting anomalies into ShapeNet models. (3) **Real3D-AD** [21] is a high-resolution industrial dataset with 1,254 samples across 12 categories, where anomalies vary significantly in shape and scale.

Comparison Methods. Nature3D-AD adopts the multi-category anomaly detection setting, where a single model is trained for all categories. All compared methods are trained under the same setting. We compare with a range of representative methods, including BTF [9], M3DM [28], PatchCore [26], CPMF [2], Reg3D-AD [21], R3D-AD [37], PO3AD [32], MC3D-AD [3], ISMP [16], Reg2Inv [34], PASDF [36], MulSen-AD [14], Simple3D [6], and SeDiR [11].

5.2 Results on NaturalGrowth

Table 1 presents the comparison results on NaturalGrowth. Our method achieves the best average O-AUROC of 0.853, surpassing the second-best method by 6.1%, and ranks first in mean rank. Methods designed for geometrically regular industrial objects, such as PASDF and R3D-AD, struggle on this dataset due to the high intra-class shape variability of agricultural products.

5.3 Results on Existing Datasets

Results on Real3D-AD. Table 2 presents the comparison results on Real3D-AD. Similarly, our method with multi-category unified training achieves the best average O-AUROC of 0.887, which is 7.7% higher than the second-best method, and also ranks first in mean rank.

Results on Anomaly-ShapeNet. Table 3 presents the results on Anomaly-ShapeNet. We directly cite results from the original papers, where most methods are trained per category. Despite using a single unified model for all 40 categories, our method achieves the best average O-AUROC of 0.957, surpassing the second-best by 2.4%, and ranks first in mean rank.

The results on both existing datasets demonstrate that our approach not only generalizes well across different data domains, but also outperforms most category-specific models with a single unified model.

Table 4: Ablation study on NaturalGrowth and Anomaly-ShapeNet datasets. We evaluate the contribution of each proposed module by progressively adding them to the baseline.

Variant	ASPE	GAA		NaturalGrowth		Anomaly-ShapeNet	
		Geo.	Spec.	O-AUROC ↑	P-AUROC ↑	O-AUROC ↑	P-AUROC ↑
(a)	×	×	×	0.792	0.781	0.891	0.823
(b)	✓	×	×	0.813	0.800	0.912	0.843
(c)	×	✓	×	0.804	0.792	0.924	0.856
(d)	×	×	✓	0.824	0.811	0.908	0.839
(e)	✓	✓	×	0.826	0.813	0.938	0.868
(f)	✓	×	✓	0.839	0.824	0.929	0.862
(g)	×	✓	✓	0.836	0.821	0.941	0.872
(h)	✓	✓	✓	**0.853**	**0.835**	**0.957**	**0.885**

5.4 Ablation Study

To validate the effectiveness of each proposed module, we conduct ablation studies on NaturalGrowth and Anomaly-ShapeNet, as shown in Table 4. Starting from the baseline (a), we progressively add each module. Each individual module consistently improves over the baseline on both datasets. Notably, the spatial and spectral branches exhibit different strengths across datasets, reflecting the distinct geometric characteristics of agricultural products and industrial parts. The full model (h) achieves the best results on both datasets, confirming the complementarity of all components.

5.5 Efficiency Analysis

We evaluate the single-sample inference efficiency on a single RTX 4090 with a batch size of 1. The model has 33.32M parameters, of which 11.22M are trainable, and a forward pass costs 234.3 GFLOPs. As shown in Table 5, when the number of input points increases from 11k to 434k, the latency only rises from 30.2 ms to 73.3 ms, and the peak inference memory stays below 1 GB. The model runs at 13.7 to 33.2 FPS, which meets the requirements of automated agricultural sorting.

We further report the latency of each stage. Since FPS samples any input into a fixed number of group centers and both ASPE and GAA operate on these tokens, the latency of the token encoder, the ViT blocks, and the reconstruction network remains unchanged as the number of points grows. The increase in latency comes entirely from the grouping stage, which is a preprocessing step shared by all methods based on group tokens.

Table 5: Efficiency analysis of Nature3D-AD on a single RTX 4090. We report the inference latency, FPS, and peak memory under different numbers of input points, together with the latency of each stage at the smallest and largest input sizes.

Points	Latency/ms ↓	FPS ↑	Memory/MB ↓	Stage	11,047 points	434,067 points
11,047	30.2	33.2	907	Grouping	6.4	47.8
50,000	47.5	21.1	907	Token encoder	5.8	5.7
100,000	58.5	17.1	908	ViT blocks	2.0	2.0
200,000	60.5	16.5	909	Reconstruction	16.2	15.8
434,067	73.3	13.7	998			

5.6 Visualization

Figure 5 visualizes the point-level anomaly score maps of different methods on NaturalGrowth categories. It can be observed that Reg3D-AD [21] and MC3D-AD [3] tend to activate over broader regions, while Simple3D [6] yields relatively conservative responses. Among all methods, ours produces anomaly score maps most consistent with the ground truth, with concentrated high responses on anomalous regions and suppressed activations on normal surfaces.

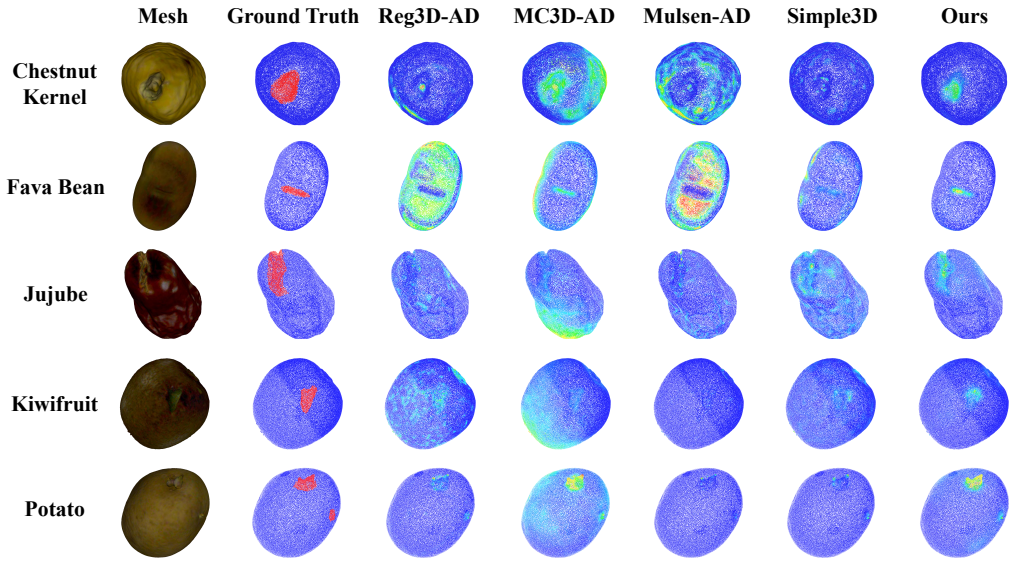


Figure 5: **Qualitative comparison of point-level anomaly localization on NaturalGrowth categories.** Warmer colors indicate higher anomaly scores.

6 Conclusion

In this paper, we advance agricultural 3D point cloud anomaly detection from both data and method perspectives. On the data side, we introduce NaturalGrowth, the first dedicated dataset comprising 31 agricultural categories, 1,477 real-world scanned samples, and 14 fine-grained anomaly types, filling the data gap in this field. On the method side, we propose Nature3D-AD, which incorporates Anomaly-Sensitive Positional Encoding (ASPE) and Geometry-Aware Attention (GAA). ASPE provides differentiated positional representations for regions of varying geometric complexity based on local curvature and density. GAA enhances the geometric awareness of self-attention from both spatial and spectral perspectives. Extensive experiments on NaturalGrowth, Anomaly-ShapeNet, and Real3D-AD demonstrate that Nature3D-AD achieves state-of-the-art performance in both anomaly detection and localization. We hope NaturalGrowth will facilitate future research on 3D anomaly detection in agricultural scenarios.

Limitation. While effective, the current pseudo-anomaly generation strategy is domain-agnostic and does not yet exploit the specific defect characteristics of agricultural products. Exploring anomaly simulation methods that better reflect real-world agricultural defect patterns could further benefit detection performance, which we leave as a promising direction for future work.

Acknowledgments and Disclosure of Funding

This research was supported partially by the Key R&D Program of Hubei Province under grant No. 2026BEB005.

References

[1] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2022.

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

[3] Jiayi Cheng, Can Gao, Jie Zhou, Jiajun Wen, Tao Dai, and Jinbao Wang. Mc3d-ad: a unified geometry-aware reconstruction model for multi-category 3d anomaly detection. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 837–845, 2025.

[4] Yuqi Cheng, Yunkang Cao, Dongfang Wang, Weiming Shen, and Wenlong Li. Boosting global-local feature matching via anomaly synthesis for multi-class point cloud anomaly detection. *IEEE Transactions on Automation Science and Engineering*, 2025.

[5] Yuqi Cheng, Yunkang Cao, Guoyang Xie, Zhichao Lu, and Weiming Shen. Toward zero-shot point cloud anomaly detection: A multiview projection framework. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2025.

[6] Yuqi Cheng, Yihan Sun, Hui Zhang, Weiming Shen, and Yunkang Cao. Towards high-resolution 3d anomaly detection: A scalable dataset and real-time framework for subtle industrial defects. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 3327–3334, 2026.

[7] Lei Fan, Dongdong Fan, Zhiguang Hu, Yiwen Ding, Donglin Di, Kai Yi, Maurice Pagnucco, and Yang Song. Manta: A large-scale multi-view and visual-text anomaly detection dataset for tiny objects. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25518–25527. IEEE, 2025.

[8] Bin-Bin Gao. Learning to detect multi-class anomalies with just one normal image prompt. In *European Conference on Computer Vision*, pages 454–470. Springer, 2024.

[9] Eliahu Horwitz and Yedid 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*, pages 2968–2977, 2023.

[10] Ermo Hua, Che Jiang, Xingtai Lv, Kaiyan Zhang, Youbang Sun, Yuchen Fan, Xuekai Zhu, Biqing Qi, Ning Ding, and Bowen Zhou. Fourier position embedding: Enhancing attention's periodic extension for length generalization. In *International Conference on Machine Learning*, pages 24932–24949, 2025.

[11] SuYeon Kim, Wongyu Lee, and MyeongAh Cho. A semantically disentangled unified model for multi-category 3d anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[12] Mathis Kruse, Marco Rudolph, Dominik Woiwode, and Bodo Rosenhahn. Splatpose & detect: Pose-agnostic 3d anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3950–3960, 2024.

[13] 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, June 2024.

[14] Wenqiao Li, Bozhong Zheng, Xiaohao Xu, Jinye Gan, Fading Lu, Xiang Li, Na Ni, Zheng Tian, Xiaonan Huang, Shenghua Gao, et al. Multi-sensor object anomaly detection: Unifying appearance, geometry, and internal properties. In *Proceedings of the computer vision and pattern recognition conference*, pages 9984–9993, 2025.

[15] Zhenye Li, Dongyi Wang, Tingting Zhu, Yang Tao, and Chao Ni. Review of deep learning-based methods for non-destructive evaluation of agricultural products. *Biosystems Engineering*, 245:56–83, 2024.

[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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5146–5154, 2025.

[17] Hanzhe Liang, Jie Zhou, Can Gao, Bingyang Guo, Jinbao Wang, and Linlin Shen. A lightweight 3d anomaly detection method with rotationally invariant features. *Pattern Recognition*, page 112924, 2025.

[18] Hanzhe Liang, Can Gao, Zehai Niu, Jie Zhang, Jinbao Wang, and Linlin Shen. An information-aware reconstruction model for multi-category 3d anomaly detection. *IEEE Transactions on Artificial Intelligence*, 2026.

[19] Hanzhe Liang, Luocheng Zhang, Junyang Xia, HanLiang Zhou, Bingyang Guo, Yingxi Xie, Can Gao, Ruiyun Yu, Jinbao Wang, and Pan Li. Open-set supervised 3d anomaly detection: An industrial dataset and a generalisable framework for unknown defects. *arXiv preprint arXiv:2604.01171*, 2026.

[20] Yuxuan Lin, Yang Chang, Xuan Tong, Jiawen Yu, Antonio Liotta, Guofan Huang, Wei Song, Deyu Zeng, Zongze Wu, Yan Wang, et al. A survey on rgb, 3d, and multimodal approaches for unsupervised industrial image anomaly detection. *Information Fusion*, 121:103139, 2025.

[21] 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.

[22] Yi Liu, Changsheng Zhang, and Yufei Yang. Template3d-ad: Point cloud template matching method based on center points for 3d anomaly detection. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 1630–1638, 2025.

[23] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pages 7176–7185, 2020.

[24] Jian Ning, Qin Zou, Linchun Wu, Yuanhao Yue, Kunmo Li, Shoubin Chen, and Zhongyuan Wang. Physics-inspired pseudo anomaly generation and prototype feature guidance for 3d anomaly detection. *Pattern Recognition*, page 114391, 2026.

[25] Yatian Pang, Eng Hock Francis Tay, Li Yuan, and Zhenghua Chen. Masked autoencoders for 3d point cloud self-supervised learning. *World Scientific Annual Review of Artificial Intelligence*, 1:2440001, 2023.

[26] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2022.

[27] Xian Tao, Dapeng Zhang, Wenzhi Ma, Zhanxin Hou, ZhenFeng Lu, and Chandranath Adak. Unsupervised anomaly detection for surface defects with dual-siamese network. *IEEE Transactions on Industrial Informatics*, 18(11):7707–7717, 2022.

[28] 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.

[29] Zeqi Xiao, Wenwei Zhang, Tai Wang, Chen Change Loy, Dahua Lin, and Jiangmiao Pang. Position-guided point cloud panoptic segmentation transformer. *International Journal of Computer Vision*, 133(1):275–290, 2025.

[30] Guoyang Xie, Jinbao Wang, Jiaqi Liu, Jiayi Lyu, Yong Liu, Chengjie Wang, Feng Zheng, and Yaochu Jin. Im-iad: Industrial image anomaly detection benchmark in manufacturing. *IEEE Transactions on Cybernetics*, 54(5):2720–2733, 2024.

[31] Kaichen Yang, Junjie Cao, Zeyu Bai, Zhixun Su, and Andrea Tagliasacchi. Piad: Pose and illumination agnostic anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4734–4743, 2025.

[32] 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.

[33] Jiangyong Yu, Changyong Shu, Sifan Zhou, Zichen Yu, Xing Hu, and Dawei Yang. Fq-petr: Fully quantized position embedding transformation for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 12144–12152, 2026.

[34] Yuyang Yu, Zhengwei Chen, Xuemiao Xu, Lei Zhang, Haoxin Yang, Yongwei Nie, and Shengfeng He. Registration is a powerful rotation-invariance learner for 3d anomaly detection. *arXiv preprint arXiv:2510.16865*, 2025.

[35] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. Destseg: Segmentation guided denoising student-teacher for anomaly detection. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3914–3923. IEEE, 2023.

[36] Bozhong Zheng, Jinye Gan, Xiaohao Xu, Xintao Chen, Wenqiao Li, Xiaonan Huang, Na Ni, and Yingna Wu. Bridging 3d anomaly localization and repair via high-quality continuous geometric representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27063–27072, 2025.

[37] 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.

[38] 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*, MM '24, page 4680–4689. ACM, October 2024.

[39] Xiangjian Zhu, Ling Ren, Miao Yang, Yuquan Zhang, Conghua Zhang, Jianpu Cui, and Junping Liu. A systematic review of deep learning-based object detection methods for crop grading. *CABI Agriculture and Bioscience*, 6(1):0060, 2025.

[40] Qin Zou, Zheng Zhang, Qingquan Li, Xianbiao Qi, Qian Wang, and Song Wang. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE transactions on image processing*, 28(3): 1498–1512, 2018.

Appendix

In this appendix, Appendix A provides additional details about the NaturalGrowth dataset together with an analysis of how well the training samples cover the normal morphology, and Appendix B provides the implementation details and supplementary quantitative and qualitative results.

A Dataset Details

A.1 Comparison with Existing Datasets

Table 6: Comparison of NaturalGrowth with existing anomaly detection datasets. 'Std.' measures the standard deviation of anomaly area ratios across all defective samples, reflecting the diversity of anomaly scales.

Dataset	Domain	#Samples	#Anomaly Types	Real Captured	No Std. Reference	Std.(%)
MVTec 3D-AD [1]	Industrial	3,853	8	✓	×	0.55
Anomaly-ShapeNet [13]	Industrial	1,471	7	×	×	2.45
Real3D-AD [21]	Industrial	1,252	3	✓	×	2.58
MulSen-AD [14]	Industrial	2,035	15	✓	×	17.49
MiniShift [6]	Industrial	2,577	2	×	×	2.25
NaturalGrowth (Ours)	Agricultural	1,477	14	✓	✓	4.36

Table 6 compares NaturalGrowth with existing 3D anomaly detection datasets across multiple dimensions. NaturalGrowth is the first dataset targeting the agricultural domain, featuring real-world captured samples without standard references, and exhibits moderate anomaly scale diversity.

A.2 Category Statistics

Table 7: Detailed data distribution for each category in NaturalGrowth.

Category	#Training	#Testing		Total	Avg. Points
	Normal	*Normal*	*Anomaly*		
Almonds	4	16	27	47	13,778
Banana	4	16	24	44	413,995
Broccoli	4	15	22	41	172,830
Carrot	4	18	51	73	152,953
Cashew	4	16	41	61	26,280
Chestnut Kernel	4	13	20	37	59,112
Chili	4	19	52	75	45,125
Clementine	4	10	20	34	153,655
Fava Bean	4	16	53	73	20,549
Green Bean	4	20	51	75	86,894
Hawthorn	4	15	20	39	86,156
Ipomoea Batatas	4	16	42	62	159,388
Jujube	4	18	27	49	100,099
Kiwifruit	4	15	15	34	186,302
Loquat	4	15	15	34	153,068
Lychee	4	16	17	37	93,922
Mandarin	4	12	30	46	433,156
Mango	4	11	13	28	349,459
Mangosteen	4	6	19	29	195,803
Mushroom	4	18	18	40	117,987
Onion	4	11	21	36	328,420
Orange	4	16	26	46	235,957
Peanut	4	16	20	40	37,915
Peanut Kernel	4	16	18	38	11,041
Persimmon	4	15	17	36	358,891
Potato	4	12	29	45	180,504
Radish	4	17	41	62	90,949
Snow Pea	4	18	51	73	112,264
Strawberry	4	15	16	35	162,493
Walnut	4	16	17	37	71,517
Yam Bean	4	19	48	71	14,267
Total	**124**	**472**	**881**	**1,477**	**135,163**

Table 7 presents the detailed data distribution for each category in NaturalGrowth, including the number of training and testing samples, and the average point cloud size. Each category has 4 normal samples for training.

The average number of points per sample varies significantly across categories, ranging from around 11,000 to over 430,000, reflecting the diverse scales of agricultural products.

A.3 Coverage of Normal Morphology by Training Samples

Since agricultural products admit no canonical shape, no finite training set can enumerate the morphologies a category may take. We therefore quantify how well the four normal training samples per category represent the normal distribution. The answer depends on how a sample is regarded. Taken as a whole, four samples are four shapes; taken as a surface composed of local neighborhoods, the same four samples provide several thousand of them, and the distribution being sampled is that of local geometry. We measure both.

We take k normal samples of a category as a reference set and, for every held-out normal sample, average the Euclidean distance from each of its descriptors to the nearest descriptor of the reference set, so that a smaller value indicates closer proximity. The global level treats each sample as a single descriptor, while the local level uses the same group neighborhoods as our model (1024 groups of 128 nearest neighbors). Both levels are evaluated in two feature spaces, the hand-crafted FPFH descriptor and the frozen PointMAE encoder. For $k > 4$, the additional reference samples are drawn from the normal test split and the evaluation is performed on the remaining ones; this analysis characterizes the data and leaves training untouched.

Table 8: Mean nearest-neighbor distance from held-out normal samples to a reference set of k normal samples (lower is closer), averaged over all categories of NaturalGrowth.

Feature space	**Level**	$k=1$	$k=2$	$k=4$	$k=8$	$k=12$
FPFH	Global	0.961	0.892	0.785	0.702	0.665
	Local	0.395	0.349	0.314	0.285	0.271
PointMAE	Global	0.985	0.833	0.634	0.501	0.446
	Local	0.192	0.156	0.120	0.116	0.114

As shown in Table 8, in both feature spaces the local distance at $k = 4$ is a fraction of the global one, and it barely decreases as further samples are added. Locally, an unseen normal sample is thus already close to what four samples provide.

The distances indicate how close an unseen sample is; we further measure what proportion of it is accounted for. We compute the fraction of held-out normal descriptors that have a training descriptor within a given radius, under two radii. The first follows Naeem et al. [23], setting the radius to the distance from a held-out descriptor to its fifth nearest neighbor, so that it adapts to the local density of the data. The second is stricter and is calibrated per category such that 95% of the annotated defective neighborhoods fall outside the covered region, where a neighborhood is regarded as defective when over 20% of its points are labeled anomalous.

Table 9: Percentage of local descriptors of held-out normal samples covered by a reference set of k normal samples (higher is better), averaged over all categories of NaturalGrowth.

Radius	**Feature space**	$k=1$	$k=2$	$k=4$	$k=8$	$k=12$
Density-based [23]	FPFH	31.4	52.7	70.6	73.0	73.6
	PointMAE	48.5	74.6	91.3	93.8	94.5
Defect-calibrated	FPFH	21.0	36.4	49.8	51.8	52.4
	PointMAE	54.2	62.8	67.9	68.7	69.0

As shown in Table 9, four samples cover 70.6% and 91.3% of the local geometry of an unseen normal sample in the two feature spaces, while the eight further samples add only 3.0 and 3.2 points. Under the stricter radius, which admits almost no defective geometry, four samples still cover 49.8% and 67.9%, and the remaining eight again add only 2.6 and 1.1 points. The four measurements differ in the absolute level they report, as they use different descriptors and radii, but all of them place the great majority of the achievable coverage within the first four samples. The four-sample protocol of NaturalGrowth therefore asks a model to learn the local features of normal geometry from a limited number of samples and to generalize from them, rather than to enumerate all morphologies a category can take.

A.4 Dataset Visualization

Figure 6 visualizes one representative normal sample from each of the 31 categories in NaturalGrowth, illustrating the wide morphological diversity of the dataset.

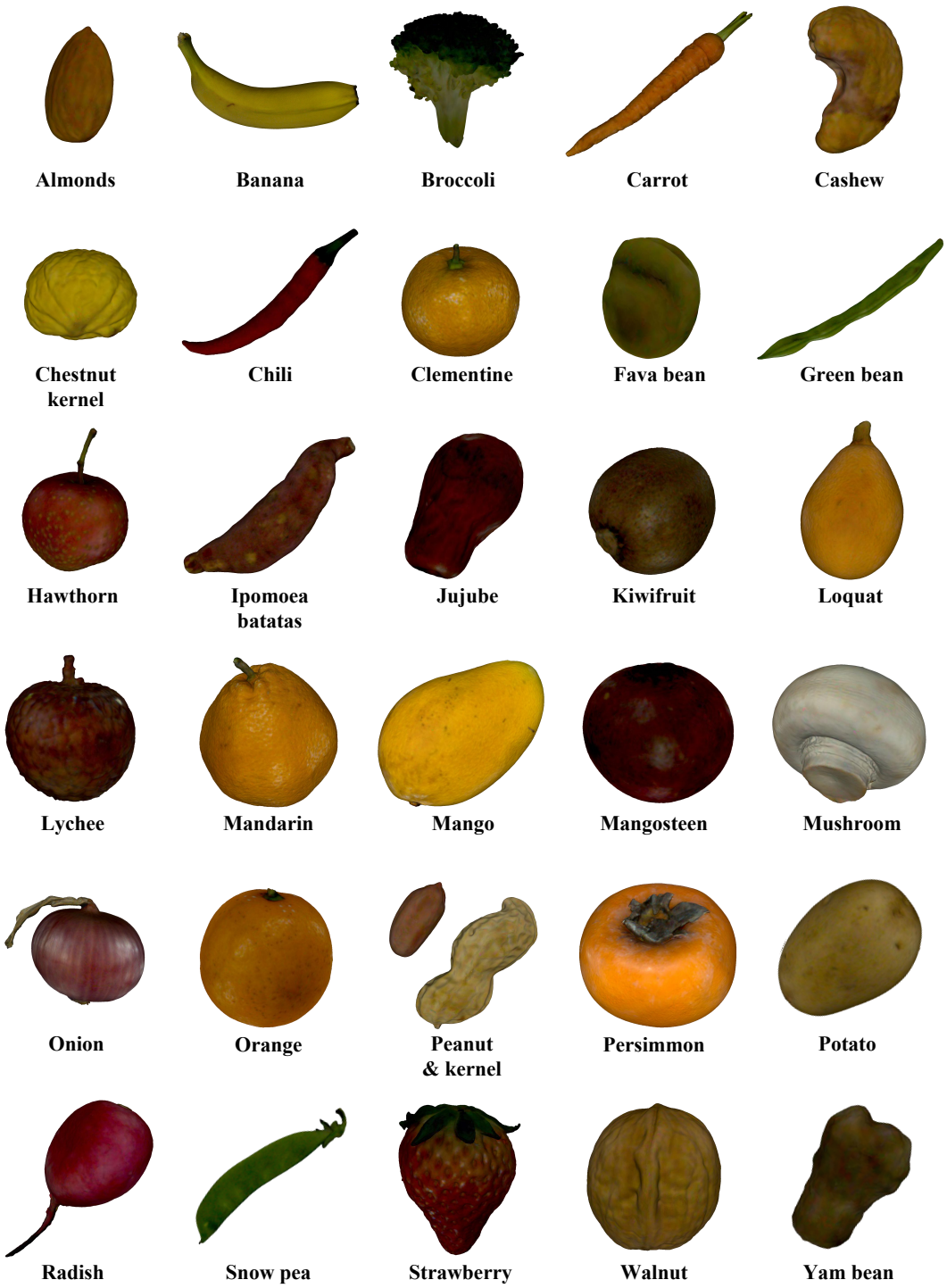


Figure 6: **Visualization of all 31 categories in the NaturalGrowth dataset.** The dataset spans a wide range of agricultural products including fruits, vegetables, nuts, and legumes, exhibiting significant diversity in shape, size, and surface texture.

Table 10: P-AUROC performance of different methods on Naturalgrowth across 31 categories, where best and second-place results are highlighted in **red** and **blue**, respectively.

P-AUROC											
Method	**almonds**	**banana**	**broccoli**	**carrot**	**cashew**	**radish**	**chestnut**	**chili**	**clementine**	**fava bean**	**green bean**
BTF(Raw) (CVPR23')	0.444	0.488	0.433	0.437	0.440	0.450	0.586	0.408	0.665	0.484	0.565
BTF(FPFH) (CVPR23')	0.616	**0.910**	0.436	0.665	0.710	0.855	0.822	0.726	0.907	0.858	0.849
M3DM (CVPR23')	0.592	0.320	0.410	0.542	0.519	0.580	0.838	0.632	0.786	0.663	0.503
PatchCore(FPFH) (CVPR22')	**0.850**	0.657	0.484	0.637	0.807	**0.857**	0.810	**0.800**	0.880	**0.954**	0.854
PatchCore(PointMAE) (CVPR22')	0.647	0.485	0.503	0.629	0.725	0.605	0.770	0.704	0.744	0.639	0.505
CPMF (PR24')	**0.884**	0.597	0.518	0.557	**0.901**	0.389	0.493	0.743	0.552	0.931	0.654
Reg3D-AD (NeurIPS23')	0.632	0.652	0.353	0.543	0.743	0.589	0.712	0.716	0.713	0.594	0.538
R3D-AD (ECCV24')	0.456	0.359	**0.921**	0.276	0.441	0.525	0.596	0.463	0.510	0.579	0.402
PO3AD (CVPR25')	0.709	0.650	0.623	0.570	0.738	0.738	0.750	0.653	0.866	0.901	0.675
MC3D-AD (IJCAI25')	0.819	0.699	0.552	0.608	0.842	0.846	**0.921**	0.779	0.924	**0.990**	0.835
ISMP (AAAI25')	0.621	0.841	0.485	0.592	0.618	0.784	0.700	0.734	0.787	0.710	0.748
Reg2Inv (NeurIPS25')	0.792	0.842	0.511	0.636	0.782	0.713	0.772	0.542	0.903	0.759	0.514
PASDF (ICCV25')	0.432	0.438	**0.630**	0.422	0.539	0.401	0.327	0.711	0.524	0.499	0.431
Mulsen-AD (CVPR25')	0.601	0.396	0.472	0.500	0.509	0.549	0.754	0.682	0.635	0.616	0.560
Simple3D (AAAI26')	0.811	**0.956**	0.487	**0.666**	0.769	0.856	0.799	0.775	**0.927**	0.881	**0.872**
Ours	0.592	0.820	0.336	**0.822**	**0.898**	**0.882**	**0.961**	**0.836**	**0.948**	0.906	**0.913**
Method	**hawthorn**	**ipomoea**	**jujube**	**kiwifruit**	**loquat**	**lychee**	**mandarin**	**mango**	**mangosteen**	**mushroom**	**onion**
BTF(Raw) (CVPR23')	0.701	0.489	0.504	0.496	0.612	0.644	0.635	0.633	0.540	0.539	0.511
BTF(FPFH) (CVPR23')	0.747	0.828	**0.908**	0.864	0.920	0.803	0.881	0.607	0.816	0.757	0.723
M3DM (CVPR23')	0.787	0.817	0.713	0.589	0.747	0.737	0.704	0.654	0.660	0.547	0.459
PatchCore(FPFH) (CVPR22')	0.762	0.746	0.871	0.854	0.876	0.764	0.888	0.540	0.774	0.746	0.596
PatchCore(PointMAE) (CVPR22')	0.611	0.532	0.642	0.640	0.701	0.807	0.617	0.506	0.672	0.565	0.586
CPMF (PR24')	0.522	0.512	0.554	0.438	0.624	0.464	0.507	0.706	0.554	0.425	0.504
Reg3D-AD (NeurIPS23')	0.691	0.592	0.669	0.568	0.707	0.811	0.662	0.542	0.623	0.530	0.553
R3D-AD (ECCV24')	0.287	0.376	0.190	0.306	0.342	0.343	0.556	0.407	0.681	0.401	0.465
PO3AD (CVPR25')	0.800	0.743	0.857	0.881	0.940	**0.853**	0.763	0.784	0.796	0.757	0.722
MC3D-AD (IJCAI25')	0.751	**0.875**	0.815	0.784	**0.964**	0.819	0.927	0.613	0.757	0.613	**0.759**
ISMP (AAAI25')	0.700	0.745	0.757	0.743	0.847	0.585	0.820	0.534	0.757	0.689	0.684
Reg2Inv (NeurIPS25')	**0.939**	0.719	0.763	0.833	0.921	0.811	**0.930**	**0.832**	**0.875**	**0.851**	0.749
PASDF (ICCV25')	0.307	0.312	0.457	0.247	0.364	0.439	0.472	0.470	0.500	0.489	0.409
Mulsen-AD (CVPR25')	0.723	0.570	0.616	0.612	0.653	0.776	0.577	0.609	0.548	0.510	0.518
Simple3D (AAAI26')	0.750	0.794	0.874	**0.935**	**0.956**	0.781	**0.975**	0.587	0.860	0.742	0.757
Ours	**0.818**	**0.908**	**0.932**	**0.937**	0.931	**0.865**	0.910	**0.928**	**0.911**	**0.903**	**0.793**
Method	**orange**	**peanut**	**peanut kernel**	**persimmon**	**potato**	**snow pea**	**strawberry**	**walnut**	**yam bean**	**Average**	**Mean Rank**
BTF(Raw) (CVPR23')	0.657	0.617	0.458	0.479	0.487	0.448	0.408	0.545	0.587	0.529	13.065
BTF(FPFH) (CVPR23')	0.941	0.643	0.684	0.847	**0.923**	0.772	0.507	0.662	0.843	0.775	5.855
M3DM (CVPR23')	0.738	0.746	0.641	0.566	0.564	0.463	**0.598**	0.573	0.826	0.629	10.081
PatchCore(FPFH) (CVPR22')	0.864	**0.824**	**0.929**	0.860	0.856	0.795	0.489	0.681	**0.933**	0.782	5.161
PatchCore(PointMAE) (CVPR22')	0.689	0.606	0.762	0.550	0.603	0.508	0.580	0.648	0.669	0.627	9.968
CPMF (PR24')	0.597	0.730	**0.911**	0.646	0.666	0.720	0.510	0.558	**0.956**	0.623	9.484
Reg3D-AD (NeurIPS23')	0.734	0.625	0.739	0.559	0.596	0.582	0.497	0.683	0.690	0.627	9.984
R3D-AD (ECCV24')	0.476	0.170	0.366	0.591	0.368	0.573	0.338	0.397	0.543	0.442	14.161
PO3AD (CVPR25')	0.836	0.573	0.842	0.612	0.840	0.507	**0.766**	0.660	0.822	0.749	6.629
MC3D-AD (IJCAI25')	**0.972**	0.676	0.833	0.891	0.836	0.682	0.595	**0.743**	0.932	0.795	4.371
ISMP (AAAI25')	0.860	0.583	0.686	0.844	0.871	0.753	0.481	0.600	0.674	0.704	8.500
Reg2Inv (NeurIPS25')	0.897	**0.833**	0.821	0.913	0.842	0.461	0.582	0.701	0.871	0.771	5.532
PASDF (ICCV25')	0.393	0.391	0.372	0.447	0.423	0.545	0.436	0.384	0.496	0.442	14.323
Mulsen-AD (CVPR25')	0.650	0.774	0.675	0.536	0.582	0.493	0.533	0.503	0.832	0.599	11.226
Simple3D (AAAI26')	0.963	0.790	0.890	**0.919**	0.916	**0.811**	0.483	0.690	0.908	**0.812**	**4.194**
Ours	**0.980**	0.619	0.685	**0.936**	**0.966**	**0.814**	0.395	**0.821**	0.918	**0.835**	**3.468**

B Additional Experimental Results

B.1 Implementation Details

We adopt PointMAE [25] pre-trained on ModelNet40 as the feature extractor, with its parameters frozen during training. The AdamW optimizer is used with an initial learning rate of 10^{-4}, which decays to 10^{-5} after 800 epochs. The batch size is set to 1 and the maximum number of training epochs is 1000. The reconstruction network consists of a 4-layer encoder and a 4-layer decoder Transformer with a hidden dimension of 256. The number of FPS-sampled group centers G is set to 1024 for NaturalGrowth and Anomaly-ShapeNet, and 4096 for Real3D-AD. In ASPE, the neighborhood size k is set to 20 and the number of Fourier encoding frequencies is 64. In GAA, the number of attention heads is 4 and the gating parameter β is initialized to 0. All experiments are conducted on a single NVIDIA RTX 4090 GPU, with unified training across all categories on NaturalGrowth taking approximately 26 hours.

Pseudo-Anomaly Generation. We follow PA-Gen [32] to synthesize pseudo-anomalies. Each training sample is corrupted with a probability of 0.7 and is otherwise fed to the network unchanged. To generate a pseudo-anomaly, a surface point is randomly selected as the center, and a spherical neighborhood covering approximately $1/64$ of all points is selected around it. Points in this neighborhood are displaced along their normals, with the displacement decaying linearly with distance from the center, so that the result is a smooth bump or dent rather than a discontinuity. The sign of the displacement is drawn at random, giving protrusions and concavities in equal proportion, and the maximum displacement is uniformly sampled from $[0.06, 0.12]$ in normalized coordinates. One anomalous region is generated per sample.

Evaluation Metrics. Following prior work [21, 32], we adopt four metrics for comprehensive evaluation. Object-level AUROC (O-AUROC) measures the ability to classify whether a sample contains anomalies. Point-level AUROC (P-AUROC) evaluates the accuracy of anomaly localization at each point. Object-level AUPR (O-AUPR) provides a more informative measure under class imbalance, as it is sensitive to the precision-recall trade-off. In the main paper, we report O-AUROC as the primary metric for detection performance. P-AUROC and O-AUPR results are provided in this appendix for completeness.

Table 11: O-AUPR performance of different methods on Naturalgrowth across 31 categories, where best and second-place results are highlighted in **red** and **blue**, respectively.

O-AUPR											
Method	**almonds**	**banana**	**broccoli**	**carrot**	**cashew**	**radish**	**chestnut**	**chili**	**clementine**	**fava bean**	**green bean**
BTF(Raw) (CVPR23)	0.527	0.481	0.506	0.634	0.637	0.669	0.828	0.655	0.756	0.970	0.729
BTF(FPFH) (CVPR23)	0.580	0.647	0.562	0.726	0.818	0.683	0.691	0.702	0.936	0.846	0.721
M3DM (CVPR23)	0.329	0.636	0.666	0.764	0.685	0.654	0.779	0.662	0.797	0.864	0.753
PatchCore(FPFH) (CVPR22)	0.910	0.572	0.571	0.715	**0.929**	0.680	0.841	0.729	0.871	0.983	0.806
PatchCore(PointMAE) (CVPR22)	0.861	**0.741**	0.642	0.782	0.884	0.663	0.770	0.855	0.808	**0.993**	0.709
CPMF (PR24)	0.529	0.177	0.104	0.219	0.518	0.080	0.078	0.002	0.186	0.637	0.240
Reg3D-AD (NeurIPS23)	0.785	0.652	0.536	0.809	0.780	0.738	0.718	0.778	0.871	0.989	0.753
R3D-AD (ECCV24)	0.592	0.517	**0.966**	0.661	0.784	0.732	0.634	0.770	**1.000**	0.768	0.824
PO3AD (CVPR25)	**0.915**	0.705	0.616	0.637	0.764	0.716	0.682	0.839	0.941	0.851	0.760
MC3D-AD (IJCAI25)	**0.940**	**0.742**	0.898	**0.839**	0.889	**0.917**	**0.997**	**0.881**	**1.000**	0.958	**0.859**
ISMP (AAAI25)	0.573	0.560	0.504	0.780	0.852	0.741	0.630	0.715	0.793	0.858	0.700
Reg2Inv (NeurIPS25)	0.808	0.642	0.563	0.794	0.702	0.725	0.740	0.756	0.978	0.934	0.723
PASDF (ICCV25)	0.620	0.573	**0.947**	0.738	0.740	0.707	0.606	0.825	0.667	0.765	0.714
Mulsen-AD (CVPR25)	0.504	0.665	0.681	0.754	0.625	0.624	0.776	0.681	0.824	0.858	0.743
Simple3D (AAAI26)	0.793	0.552	0.720	0.744	0.857	0.653	0.928	0.700	**0.981**	**0.994**	0.832
Ours	0.845	0.637	0.682	**0.938**	**0.964**	**0.855**	**1.000**	**0.885**	**1.000**	0.992	**0.929**

Method	**hawthorn**	**ipomoea**	**jujube**	**kiwifruit**	**loquat**	**lychee**	**mandarin**	**mango**	**mangosteen**	**mushroom**	**onion**
BTF(Raw) (CVPR23)	0.575	0.851	0.796	0.570	0.657	0.678	0.865	0.668	0.846	0.599	0.766
BTF(FPFH) (CVPR23)	0.557	0.691	0.767	0.535	0.577	0.489	0.841	0.760	0.681	0.477	0.583
M3DM (CVPR23)	0.639	0.914	0.749	0.460	0.641	0.665	0.871	0.755	0.767	0.563	0.667
PatchCore(FPFH) (CVPR22)	0.527	0.686	0.877	0.446	0.655	0.748	0.920	0.601	0.764	0.603	0.791
PatchCore(PointMAE) (CVPR22)	0.611	0.802	0.770	0.691	0.645	0.744	0.861	0.667	0.824	0.671	0.663
CPMF (PR24)	0.122	0.145	0.155	0.165	0.230	0.135	0.148	0.279	0.167	0.074	0.106
Reg3D-AD (NeurIPS23)	0.518	0.820	0.680	0.462	0.518	0.587	0.854	0.766	0.784	0.667	0.726
R3D-AD (ECCV24)	0.392	0.677	0.619	0.608	**0.793**	0.601	0.714	0.772	0.644	0.634	0.610
PO3AD (CVPR25)	0.514	0.853	0.825	**0.887**	0.765	0.715	0.750	**0.861**	0.903	0.666	0.754
MC3D-AD (IJCAI25)	**0.846**	0.912	**0.988**	0.835	0.697	0.824	0.861	0.849	**0.941**	0.858	**0.813**
ISMP (AAAI25)	0.489	0.731	0.584	0.571	0.374	0.611	0.825	0.688	0.718	0.429	0.629
Reg2Inv (NeurIPS25)	0.836	0.670	0.696	0.565	0.652	**0.848**	**0.946**	0.840	0.933	**0.951**	0.736
PASDF (ICCV25)	0.571	0.724	0.614	0.500	0.534	0.533	0.714	0.542	0.760	0.474	0.632
Mulsen-AD (CVPR25)	0.645	**0.918**	0.733	0.527	0.577	0.732	0.860	0.784	0.839	0.628	0.719
Simple3D (AAAI26)	0.643	0.799	0.961	0.687	**0.820**	0.637	0.891	0.673	0.813	0.432	0.736
Ours	**0.865**	**0.968**	**0.989**	**0.842**	0.760	**0.899**	**0.942**	**0.892**	**1.000**	**0.964**	**0.858**

Method	**orange**	**peanut**	**peanut kernel**	**persimmon**	**potato**	**snow pea**	**strawberry**	**walnut**	**yam bean**	**Average**	**Mean Rank**
BTF(Raw) (CVPR23)	0.694	**0.920**	0.627	0.699	0.763	**0.830**	0.745	0.639	0.704	0.706	9.032
BTF(FPFH) (CVPR23)	0.762	0.728	0.514	0.658	0.961	0.734	0.562	0.663	0.884	0.688	9.968
M3DM (CVPR23)	0.715	**0.853**	0.768	0.626	0.757	0.795	0.786	0.746	0.775	0.713	8.548
PatchCore(FPFH) (CVPR22)	0.728	0.659	0.704	0.598	0.874	0.729	0.533	0.792	0.903	0.734	7.919
PatchCore(PointMAE) (CVPR22)	0.758	0.654	0.800	0.566	0.770	0.734	0.679	0.690	0.735	0.743	7.629
CPMF (PR24)	0.221	0.170	0.654	0.237	0.246	0.229	0.182	0.142	0.758	0.237	15.419
Reg3D-AD (NeurIPS23)	0.684	0.541	0.612	0.562	0.780	0.701	0.726	0.723	0.695	0.704	9.323
R3D-AD (ECCV24)	0.682	0.365	0.432	0.566	0.748	0.606	0.438	0.542	0.825	0.662	10.548
PO3AD (CVPR25)	0.725	0.500	0.617	0.612	**0.977**	0.793	**0.851**	0.757	0.875	0.762	6.806
MC3D-AD (IJCAI25)	**1.000**	0.754	**0.928**	**0.781**	0.927	**0.943**	**0.928**	**0.916**	**0.998**	**0.888**	**2.710**
ISMP (AAAI25)	0.619	0.526	0.530	0.517	0.734	0.722	0.569	0.583	0.750	0.642	11.790
Reg2Inv (NeurIPS25)	0.830	0.725	0.731	0.712	0.898	0.753	0.732	0.563	0.736	0.765	7.048
PASDF (ICCV25)	0.662	0.544	0.529	0.570	0.707	0.629	0.534	0.515	0.709	0.642	12.016
Mulsen-AD (CVPR25)	0.666	0.839	0.635	0.664	0.726	0.783	0.703	0.726	0.756	0.716	8.742
Simple3D (AAAI26)	**0.931**	0.755	0.794	0.751	0.956	0.795	0.414	0.881	0.915	0.775	6.290
Ours	**1.000**	0.829	**0.969**	**0.968**	**0.985**	0.790	0.739	**0.953**	**0.997**	**0.901**	**2.210**

Table 12: P-AUROC performance of different methods on AnomalyShapeNet 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
ParchCore(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
ParchCore(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
RIF (PR25')	0.865	**0.957**	**0.952**	0.710	0.813	**0.979**	0.613	0.859	0.872	0.802	**0.927**	0.553	0.812	**0.968**
PO3AD (CVPR25')	**0.962**	0.949	0.912	0.844	0.880	0.978	**0.914**	**0.918**	**0.935**	**0.967**	**0.941**	**0.755**	**0.899**	**0.957**
Reg2Inv (NeurIPS25')	0.785	**0.991**	**0.995**	0.849	0.817	**0.983**	0.828	0.822	0.761	0.788	0.824	0.610	0.855	0.861
SeDiR (CVPR26')	0.764	0.898	0.911	**0.898**	**0.945**	0.866	0.700	0.795	0.869	0.806	0.646	0.717	0.884	0.900
Ours	**0.949**	0.922	0.928	**0.883**	**0.933**	0.889	**0.873**	**0.924**	**0.939**	**0.875**	0.873	**0.792**	**0.915**	0.934

Method	**cap3**	**cap4**	**cap5**	**cup0**	**cup1**	**eraser0**	**headset0**	**headset1**	**helmet0**	**helmet1**	**helmet2**	**helmet3**	**jar0**	**micro.0**
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
ParchCore(FPFH) (CVPR22')	0.653	0.595	0.795	0.655	0.596	0.810	0.583	0.464	0.548	0.489	0.445	0.737	0.478	0.488
ParchCore(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')	0.755	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
RIF (PR25')	0.955	**0.956**	0.943	**0.912**	0.831	**0.976**	0.895	0.840	0.838	0.639	0.905	0.719	**0.967**	**0.955**
PO3AD (CVPR25')	0.948	0.940	0.864	0.909	**0.932**	0.974	0.823	**0.907**	0.878	**0.948**	**0.932**	0.846	0.871	0.810
Reg2Inv (NeurIPS25')	0.945	0.864	**0.970**	0.798	0.881	**0.980**	**0.946**	**0.970**	**0.925**	0.906	0.891	**0.956**	**0.982**	**0.992**
SeDiR (CVPR26')	**0.972**	0.940	0.950	0.871	0.842	0.748	0.708	0.687	0.810	0.596	0.885	0.701	0.911	0.891
Ours	**0.965**	**0.948**	**0.981**	**0.920**	**0.924**	0.929	**0.897**	0.781	**0.887**	**0.909**	**0.910**	**0.877**	0.921	0.913

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	11.637
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	9.350
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	9.725
ParchCore(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	10.650
ParchCore(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	10.850
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	11.325
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	8.062
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	8.575
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	0.690	7.612
RIF (PR25')	0.765	0.670	0.667	0.905	0.807	0.930	0.863	0.842	0.696	0.913	0.938	0.775	0.845	4.037
PO3AD (CVPR25')	0.663	0.783	0.692	**0.955**	**0.882**	**0.978**	**0.884**	0.902	**0.937**	**0.982**	**0.950**	**0.952**	**0.899**	**2.763**
Reg2Inv (NeurIPS25')	0.632	**0.918**	**0.869**	**0.980**	0.705	**0.997**	0.844	**0.927**	0.879	0.863	0.934	**0.971**	0.882	3.150
SeDiR (CVPR26')	0.679	0.597	0.630	0.846	0.722	0.850	0.830	0.884	0.766	0.777	0.900	0.835	0.810	5.013
Ours	**0.828**	0.754	**0.881**	0.914	**0.892**	0.887	**0.895**	**0.933**	**0.885**	**0.944**	**0.946**	0.935	**0.902**	**2.250**

B.2 Supplementary Quantitative Results

Table 10 and Table 11 report the per-category P-AUROC and O-AUPR results on NaturalGrowth, respectively. Table 12 and Table 13 report the per-category P-AUROC and O-AUPR results on AnomalyShapeNet, respectively. Nature3D-AD consistently achieves the best or second-best performance across the majority of categories on both datasets, further validating the effectiveness of our method.

Table 13: O-AUPR performance of different methods on AnomalyShapeNet across 40 categories, where best and second-place results are highlighted in **red** and **blue**, respectively.

	O-AUPR													
Method	**ashtray0**	**bag0**	**bottle0**	**bottle1**	**bottle3**	**bowl0**	**bowl1**	**bowl2**	**bowl3**	**bowl4**	**bowl5**	**bucket0**	**bucket1**	**cap0**
BTF(Raw) (CVPR23')	0.578	0.458	0.466	0.573	0.543	0.588	0.464	0.576	0.654	0.601	0.615	0.652	0.620	0.659
BTF(FPFH) (CVPR23')	0.651	0.551	0.644	0.625	0.602	0.576	0.648	0.515	0.499	0.632	0.699	0.483	0.648	0.618
M3DM (CVPR23')	0.632	0.642	0.763	0.674	0.451	0.525	0.515	0.630	0.635	0.571	0.601	0.609	0.507	0.564
ParchCore(FPFH) (CVPR22')	0.445	0.608	0.615	0.677	0.579	0.548	0.545	0.611	0.620	0.575	0.541	0.604	0.565	0.585
ParchCore(PointMAE) (CVPR22')	0.679	0.601	0.545	0.645	0.651	0.562	0.611	0.456	0.556	0.601	0.585	0.541	0.642	0.561
CPMF (PR24')	0.453	0.655	0.588	0.592	0.505	0.775	0.621	0.601	0.418	0.683	0.685	0.662	0.501	0.601
Reg3D-AD (NeurIPS23')	0.588	0.608	0.632	0.695	0.474	0.494	0.515	0.495	0.441	0.624	0.555	0.632	0.714	0.693
IMRNet (CVPR24')	0.612	0.665	0.558	0.702	0.648	0.481	0.504	0.681	0.614	0.630	0.652	0.578	0.732	0.711
PO3AD (CVPR25')	**0.999**	**0.809**	**0.927**	**0.959**	**0.962**	**0.946**	**0.905**	**0.888**	**0.927**	**0.985**	**0.904**	**0.923**	**0.882**	**0.841**
Ours	**0.990**	**0.852**	**0.921**	**0.945**	**0.941**	**0.952**	**0.913**	**0.925**	**0.957**	**0.997**	**0.917**	**0.941**	**0.902**	**0.862**
Method	**cap3**	**cap4**	**cap5**	**cup0**	**cup1**	**eraser0**	**headset0**	**headset1**	**helmet0**	**helmet1**	**helmet2**	**helmet3**	**jar0**	**micro.0**
BTF(Raw) (CVPR23')	0.612	0.515	0.653	0.601	0.701	0.425	0.379	0.515	0.559	0.388	0.615	0.526	0.428	0.613
BTF(FPFH) (CVPR23')	0.579	0.545	0.593	0.585	0.651	0.719	0.531	0.523	0.568	0.721	0.588	0.564	0.479	0.662
M3DM (CVPR23')	0.652	0.477	0.642	0.570	0.752	0.625	0.632	0.623	0.528	0.627	0.636	0.458	0.555	0.464
ParchCore(FPFH) (CVPR22')	0.457	0.655	0.725	0.604	0.586	0.584	0.701	0.601	0.525	0.630	0.475	0.494	0.499	0.332
ParchCore(PointMAE) (CVPR22')	0.583	0.721	0.542	0.642	0.710	0.801	0.515	0.423	0.633	0.571	0.496	0.611	0.463	0.652
CPMF (PR24')	0.541	0.645	0.697	0.647	0.609	0.544	0.602	0.619	0.333	0.501	0.477	0.645	0.618	0.655
Reg3D-AD (NeurIPS23')	0.711	0.623	0.770	0.531	0.638	0.424	0.538	0.617	0.600	0.381	0.618	0.468	0.601	0.614
IMRNet (CVPR24')	0.702	0.658	0.502	0.455	0.627	0.599	0.701	0.656	0.697	0.615	0.602	0.575	0.760	0.552
PO3AD (CVPR25')	**0.906**	**0.876**	**0.801**	**0.879**	**0.870**	**0.995**	**0.765**	**0.914**	**0.864**	**0.961**	**0.934**	**0.849**	**0.915**	**0.803**
Ours	**0.927**	**0.892**	**0.861**	**1.000**	**0.884**	**0.978**	**0.780**	**0.855**	**0.900**	**1.000**	**0.893**	**0.859**	**1.000**	**1.000**
Method	**shelf0**	**tap0**	**tap1**	**vase0**	**vase1**	**vase2**	**vase3**	**vase4**	**vase5**	**vase7**	**vase8**	**vase9**	**Average**	**Mean Rank**
BTF(Raw) (CVPR23')	0.624	0.535	0.594	0.562	0.441	0.413	0.717	0.428	0.615	0.547	0.416	0.482	0.549	7.537
BTF(FPFH) (CVPR23')	0.611	0.610	0.575	0.641	0.655	0.569	0.652	0.587	0.472	0.592	0.624	0.638	0.598	6.350
M3DM (CVPR23')	0.665	0.722	0.638	**0.788**	0.652	0.615	0.551	0.526	0.633	0.648	0.463	0.651	0.603	6.188
ParchCore(FPFH) (CVPR22')	0.504	0.712	0.684	0.645	0.623	0.801	0.481	0.777	0.515	0.621	0.515	0.660	0.588	6.713
ParchCore(PointMAE) (CVPR22')	0.543	0.712	0.542	0.548	0.572	0.711	0.455	0.586	0.585	0.652	0.655	0.634	0.595	6.650
CPMF (PR24')	**0.681**	0.639	0.697	0.632	0.645	0.632	0.588	0.655	0.518	0.432	0.673	0.618	0.597	6.175
Reg3D-AD (NeurIPS23')	0.675	0.676	0.599	0.615	0.468	0.641	0.651	0.505	0.588	0.455	0.629	0.574	0.584	6.700
IMRNet (CVPR24')	0.625	0.401	**0.796**	0.573	**0.725**	0.655	0.708	0.528	0.654	0.601	0.639	0.462	0.621	5.600
PO3AD (CVPR25')	0.680	**0.856**	0.709	0.753	**0.789**	**0.963**	**0.902**	**0.824**	**0.879**	**0.971**	**0.833**	**0.904**	**0.881**	**1.850**
Ours	**0.740**	**0.874**	**0.866**	**0.844**	**0.725**	**0.917**	**0.903**	**0.869**	**0.890**	**0.995**	**0.882**	**0.949**	**0.907**	**1.238**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our two contributions: the NaturalGrowth benchmark dataset and the Nature3D-AD detection framework with ASPE and GAA modules. All claims are supported by experimental results on three datasets.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in the Conclusion section, noting that the current pseudo-anomaly generation strategy is domain-agnostic and does not yet exploit agricultural-specific defect patterns.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper does not include theoretical results or proofs. It is an empirical study proposing a dataset and detection method.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed experimental settings including network architecture, training hyperparameters, and evaluation metrics in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the NaturalGrowth dataset and source code upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details including optimizer, learning rate, batch size, number of epochs, and data splits are provided in Section 5 and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following the common practice in 3D anomaly detection literature, we report single-run results. The deterministic nature of our method with fixed random seeds ensures reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the GPU type, training epochs, and training time in the experimental settings.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research focuses on agricultural product quality inspection and does not involve human subjects, sensitive data, or potential harmful applications.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work aims to improve agricultural quality inspection, contributing to food safety assurance and automated sorting efficiency. We do not foresee negative societal impacts from this research.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The NaturalGrowth dataset contains only 3D scans of agricultural products and poses no risk for misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cite all existing datasets and methods used in our experiments. The pre-trained PointMAE model is used under its original license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The NaturalGrowth dataset is thoroughly documented in Section 3, including data acquisition pipeline, annotation protocol, and dataset statistics. The dataset will be released under the CC-BY 4.0 license.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are not used as a component of the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.