Towards Generalizable 3D Anomaly Detection via Relational Inconsistency Modeling

KunHo Heo*
Kyung Hee University
hkh7710@khu.ac.kr

SuYeon Kim*
Kyung Hee University
spoiuy3@khu.ac.kr

Hayoung Lee
Kyung Hee University
lhayoung9@khu.ac.kr

Chanse Oh
Kyung Hee University
chanse0727@khu.ac.kr

MyeongAh Cho†
Kyung Hee University
maycho@khu.ac.kr

Abstract

3D anomaly detection (3DAD) aims to identify defective regions in point cloud data, serving as a critical component in industrial inspection systems. Existing methods are normality-centered — learning the distribution of normal samples and treating deviations as anomalies — without explicitly modeling what constitutes a defect. This leads to ambiguous decision boundaries with increased false positives and negatives, particularly in unified and cross-domain settings where diverse normal distributions further blur the boundaries. We propose a **relational inconsistency** modeling framework that characterizes defects as violations of geometric consistency among neighboring structures. Our approach learns category-agnostic defect cues through pseudo-anomalies designed as controlled relational violations, instantiated by two key modules: **Edge-aware Graph Refinement (EGR)** for encoding geometric relationships among local regions, and **Cluster-Deviation Modeling (CDM)** for identifying regions that are relationally incompatible within their structural peer group. Extensive experiments on Anomaly-ShapeNet and Real3D-AD demonstrate consistent improvements over prior state-of-the-art methods in both in-domain and cross-domain settings, validating the effectiveness of learning an explicit, relation-based defect criterion for 3D anomaly detection. Project page: https://github.com/VisualScienceLab-KHU/GRIM.

1 Introduction

3D anomaly detection (3DAD) aims to identify defective regions in point cloud data and serves as a critical component in industrial inspection and automation systems. Unlike anomaly detection in 2D images [34, 9, 38, 27], which primarily relies on appearance cues such as color and texture, defects in 3D environments frequently manifest as violations of geometric consistency among spatially neighboring structures rather than as absolute abnormalities of isolated local regions. Due to the inherent scarcity of anomalous samples in real-world industrial settings, 3DAD is typically formulated as an unsupervised problem, where a model learns the distribution of normality from defect-free data and identifies deviations as anomalies at test time [20, 44, 17, 43, 6, 19].

Prior work on 3DAD has largely evolved along two main directions: reconstruction-based methods and feature-similarity-based methods. Reconstruction-based methods train a model to reconstruct normal samples and use reconstruction error as an anomaly score, under the assumption that anomalous

* Equal contribution
† Corresponding author

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

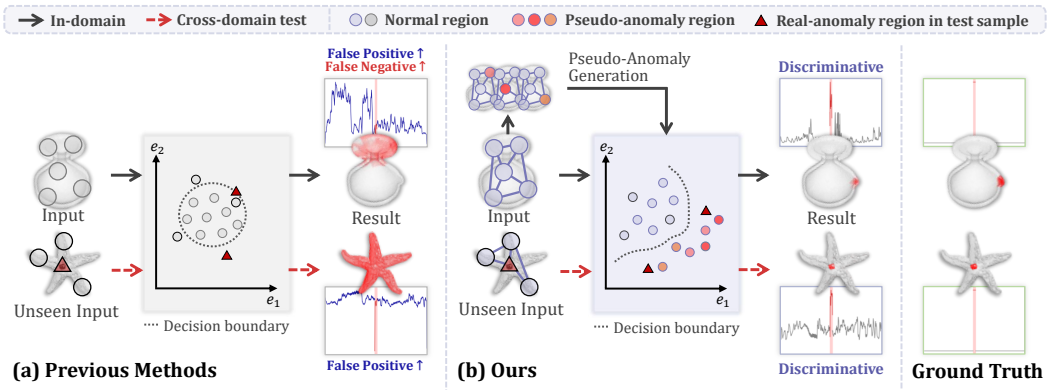


Figure 1: Comparison of previous normality-centered 3DAD methods and our relation-based defect learning framework. (a) Previous methods (e.g., MC3D-AD) rely on normal feature distributions, often resulting in false positives and false negatives. (b) Our method uses pseudo-anomalies as controlled violations of geometric relations among normal structures. By distinguishing relation-violating pseudo-anomalies from normal regions, the model produces more discriminative point-level anomaly scores, reducing both false positives and false negatives.

regions will fail to be reconstructed accurately. Feature-similarity-based methods, often implemented via memory banks, determine abnormality by measuring how dissimilar a test feature is from the distribution of normal features. While these normality-centered approaches have demonstrated effectiveness in capturing normality, they share a fundamental limitation: neither **explicitly learns a defect criterion that characterizes anomalies as relation-level inconsistencies with respect to surrounding structures**. As illustrated in Figure 1(a), such ambiguity leads to two typical failure modes: rare but valid normal regions exceed the decision threshold (false positives), while true defects remain within the normal boundary and are missed (false negatives). These failure modes indicate that deviation from the normal distribution alone is insufficient for reliable anomaly detection — without directional guidance, any such deviation may be mistaken for a true defect. This limitation becomes more pronounced in practical scenarios, where most existing methods are trained in a category-specific manner [43, 19, 33, 18] — a separate model per object category — making them ill-suited for real-world systems that require **unified (multi-category)** and **cross-domain (unseen-category)** generalization. Under such scenarios, increasingly diverse and entangled normal feature distributions further blur decision boundaries, exacerbating both false positives and false negatives.

We analyze these challenges in Figure 2 using MC3D-AD [6], a representative unified 3DAD model. The figure reports FPR at a fixed true positive rate (TPR) and FNR at a fixed true negative rate (TNR) to quantify false alarms on normal regions and missed detections of anomalies, respectively. As shown, MC3D-AD exhibits high FPR and FNR, indicating that normal regions are frequently misclassified as anomalies while true defects are often missed. This degradation is further amplified in cross-domain settings, as MC3D-AD learns category-dependent normality without explicitly learning a defect criterion based on relational inconsistencies, limiting its generalization to unseen domains.

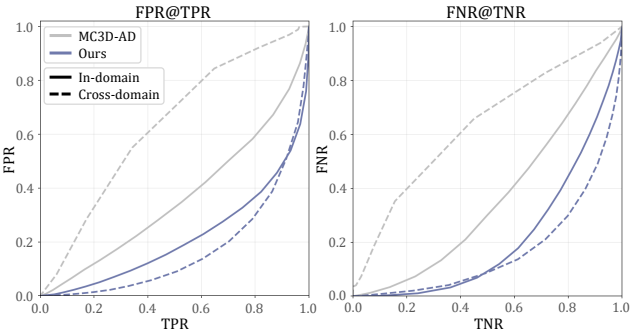


Figure 2: Comparison of False Positive Rate (FPR) and False Negative Rate (FNR) between two unified 3DAD models, MC3D-AD and our method.

Motivated by this perspective, our work aims to learn category-agnostic defect cues rather than defining diverse defect patterns individually. We argue that many industrial 3D defects are revealed less by the absolute abnormality of a local patch than by inconsistencies that break the geometric relations with surrounding structures. We utilize pseudo-anomalies for supervision, using controlled

violations of inter-structural relations to train the model to distinguish structurally coherent regions from relation-violating ones, rather than to simulate the full diversity of possible defect appearances. This formulation provides the model with an explicit criterion for defect identification grounded in **relational inconsistency**, rather than relying solely on deviation from a learned normal distribution.

To realize this idea, we propose **GRIM**, which incorporates two key designs to effectively capture the relational nature of defects. First, we represent each 3D sample as a relational graph composed of local structures and refine regional features via **Edge-aware Graph Refinement (EGR)**, which performs relation-aware message passing to encode geometric relationships among neighboring regions. This design enables the model to capture not only the absolute representation of each local region, but also the structural compatibility reflected in their relative arrangement and connectivity. Second, we introduce **Cluster-Deviation Modeling (CDM)**, which measures each region's anomaly score relative to clusters of structurally similar regions, rather than against the entire normal distribution. By clustering regional features and quantifying their deviation from cluster centers, CDM captures distribution-level inconsistencies at a finer granularity, identifying regions that are relationally incompatible within their structural peer group even when their absolute feature values appear normal. As a result, Figure 1(b) shows that our method learns an explicit defect criterion, leading to a more discriminative decision boundary and point-level anomaly scores that better separate normal and anomalous regions, thereby reducing both false positives and false negatives.

Extensive experiments validate the effectiveness of our framework, showing consistent improvements over prior state-of-the-art methods with gains of **7.4/4.7** and **7.0/3.5** in object-/point-level AUROC on Anomaly-ShapeNet and Real3D-AD, respectively. Furthermore, under cross-domain setting, our method achieves **83.6/89.6** when trained on Anomaly-ShapeNet and evaluated on Real3D-AD, and **91.7/78.9** in the reverse setting, demonstrating strong generalization in 3D anomaly detection.

Our contributions are summarized as follows:

- We revisit 3D anomaly detection by defining defects as relation-level inconsistencies, and show that normality-centered approaches lead to ambiguous decision boundaries, resulting in increased false positives and negatives, especially in unified and cross-domain settings.
- We propose a **relational inconsistency** modeling framework that learns an explicit, category-agnostic defect criterion via pseudo-anomalies, instantiated by two key modules: **Edge-aware Graph Refinement (EGR)** for relation-aware feature learning and **Cluster-Deviation Modeling (CDM)** for capturing relative distributional inconsistencies.
- We validate our approach through extensive experiments on Real3D-AD and Anomaly-ShapeNet, achieving consistent gains of **7.0/3.5** and **7.4/4.7** in object-/point-level AUROC over prior state-of-the-art methods, respectively. Our method further demonstrates strong generalization in 3D anomaly detection, particularly under cross-domain settings.

2 Related Work

2.1 Point Cloud 3D Anomaly Detection

While 2D anomaly detection has been extensively studied [12, 32, 11, 31, 10, 35, 41, 24], recent efforts have extended these ideas to 3D data to overcome blind spots in 2D representations. With the emergence of high-resolution 3D inspection benchmarks [20, 17, 2, 4], point-cloud 3D anomaly detection (3DAD) has gained increasing attention. Existing methods can be broadly categorized into feature-similarity and reconstruction-based approaches. Feature-similarity methods [20, 30, 44, 18, 3, 14, 8, 21] estimate normality by comparing test features with stored normal representations, but their reliance on memory banks limits scalability to unified settings. Reconstruction-based methods [17, 43, 6, 19] instead learn to recover normal geometry or features, using reconstruction error as anomaly scores; however, they often suffer from over-generalization, reconstructing defects while over-penalizing rare normal structures. Despite these diverse designs, most existing approaches identify anomalies as deviations from learned normality rather than explicitly modeling what constitutes a defect. This limitation becomes critical in unified and cross-domain settings, where diverse normal patterns blur decision boundaries, leading to misclassification of both normal variations and true defects. In contrast, our method learns an explicit, category-agnostic defect criterion based on relation-level inconsistencies.

2.2 Learning with Synthetic Anomalies

Synthetic anomalies are widely used to address the scarcity of real defect annotations. In 2D anomaly detection, prior work generates image- or feature-level perturbations to learn discriminative representations [16, 37, 22, 29, 40, 39, 15]. This idea has been extended to 3DAD by synthesizing pseudo-defects in point clouds. For example, R3D-AD [43] generates diverse defect shapes for reconstruction-based learning, while PO3AD [33] generates normal-vector-guided pseudo-anomaly points and learns point offsets toward normal geometry. Other methods employ geometric perturbations such as deformation or stretching to enhance representation learning [7, 13]. While effective, these approaches typically treat synthetic anomalies as proxies that mimic real defects. In contrast, our work uses pseudo-anomalies not to imitate specific defect patterns, but to provide controlled supervision for violations of geometric relations among normal structures. This allows the model to learn a category-agnostic defect criterion grounded in relational inconsistency.

3 Methods

3.1 Problem Statement

Let $\mathcal{O} = \{1, \ldots, O\}$ denote the set of object categories. For each category $o \in \mathcal{O}$, the training set $\mathcal{D}^o_{\text{train}}$ consists solely of unlabeled normal point clouds $\mathcal{P} \in \mathbb{R}^{N\times 3}$, where N denotes the number of points, following the standard anomaly detection protocol. The test set $\mathcal{D}^o_{\text{test}}$ contains both normal and anomalous samples, each annotated with an object-level binary label and point-level anomaly labels for localization. In the unified multi-category setting, a single model is trained and evaluated across all categories without category distinction, where $\mathcal{D}_{\text{train}} = \bigcup_{o=1}^{O} \mathcal{D}^o_{\text{train}}$ and $\mathcal{D}_{\text{test}} = \bigcup_{o=1}^{O} \mathcal{D}^o_{\text{test}}$ aggregate samples from all categories. Given a test point cloud, the model predicts anomaly scores over its constituent point regions for both object-level detection and point-level localization.

3.2 Pseudo-Anomaly Generation

To provide explicit supervision for learning defect criteria, we generate pseudo-anomalies from normal point clouds. The generated anomalies serve as controlled local perturbations designed to violate geometric relations among neighboring structures, encouraging the model to learn category-agnostic cues of relation-level inconsistency. Given a normal point cloud $\mathcal{P}$, the pseudo-anomaly generator $\mathcal{A}(\cdot)$ produces a corrupted point cloud $\mathcal{P}^a$ along with a point-level binary mask $\mathcal{M} = \{m_n\}_{n=1}^{N}$, where each m_n indicates whether point $\mathbf{p}_n$ has been synthetically modified, serving as point-level supervision during training. To synthesize a localized perturbation, we randomly sample a center point and construct a local patch via its nearest neighbors, confining the deformation while preserving surrounding structures.

We consider three geometric perturbation types: *bulge*, *sink*, and *hole*. Bulge and sink displace the patch along the estimated surface normal in opposite directions, producing protrusion- and depression-like distortions, respectively. Hole moves the central region toward its boundary and pushes it slightly inward while mildly expanding the boundary, forming a cavity-like deformation. These types provide complementary supervision through outward deformation, inward deformation, and local surface-support depletion. Their purpose is to expose violations of local geometric consistency, not to directly approximate the full distribution of real defects or constitute a complete defect taxonomy. Implementation details and visual examples are provided in Section 6.1, while their effectiveness as training supervision is analyzed in Section 4.4.

3.3 Edge-aware Graph Refinement

Detecting anomalies in 3D point clouds requires not only characterizing individual local regions, but also understanding their geometric relationships with neighboring structures. To this end, we propose **Edge-aware Graph Refinement (EGR)**, illustrated in Figure 3, which models each point cloud as a relational graph of local regions and performs relation-aware message passing to encode structural compatibility among neighboring nodes.

Node Features. Given an input point cloud $\mathcal{P}$ (or $\mathcal{P}^a$ in the case of pseudo-anomaly samples), we first partition it into G local groups $\{\mathcal{P}^g\}_{g=1}^{G}$ by selecting representative centers via Farthest-Point Sampling (FPS) [26] and collecting the k nearest neighbors around each center. For each group,

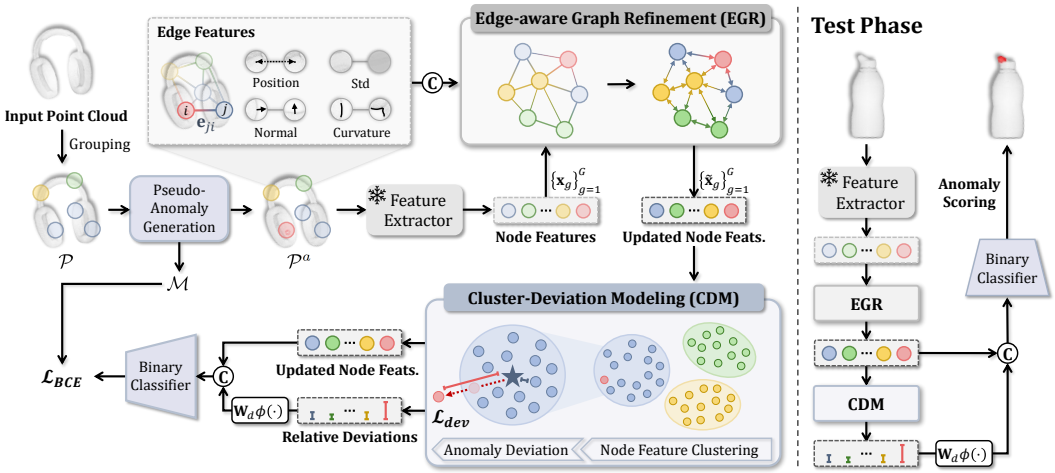


Figure 3: Overview of the proposed method. Given grouped features from an input sample as nodes, we construct edge features encoding geometric relationships between local groups, and perform Edge-aware Graph Refinement (EGR) to obtain refined node features. The refined features are then passed through Cluster-Deviation Modeling (CDM), which captures relative deviations within clusters, and fed into the classifier together with the features.

the point coordinates are normalized by subtracting the corresponding sampling center, yielding a locally translated neighborhood $\widetilde{\mathcal{P}}^g = \{\mathbf{p} - \mathbf{c}_g \mid \mathbf{p} \in \mathcal{P}^g\}$. Each normalized group is then encoded using a pretrained local encoder $E(\cdot)$ (*e.g.*, PointMAE [25]), producing a group-level feature $\mathbf{x}_g = E(\widetilde{\mathcal{P}}^g) \in \mathbb{R}^D$, where D denotes the feature dimension. The resulting features $\{\mathbf{x}_g\}_{g=1}^G$ are subsequently adopted as node features in the graph module, establishing a strict one-to-one correspondence between local point groups and graph nodes.

Edge Features. For each group $\mathcal{P}^g$, we compute a geometric descriptor comprising the group centroid $\boldsymbol{\mu}_g$, the coordinate-wise standard deviation $\boldsymbol{\sigma}_g$, the surface normal $\mathbf{n}_g$, and the curvature κ_g. The surface normal and curvature are estimated via eigendecomposition of the covariance matrix of the grouped points: the normal $\mathbf{n}_g$ is taken as the eigenvector associated with the smallest eigenvalue, and the curvature κ_g is defined as the ratio of the smallest eigenvalue to the sum of all eigenvalues. For a directed group pair (j, i), the pairwise relationship feature is defined as

$$\mathbf{e}_{ji} = \left[(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \,\|\, (\boldsymbol{\sigma}_i - \boldsymbol{\sigma}_j) \,\|\, (1 - |\mathbf{n}_i^\top \mathbf{n}_j|) \,\|\, |\kappa_i - \kappa_j| \right], \tag{1}$$

where $[\cdot \,\|\, \cdot]$ denotes feature-wise concatenation. Specifically, $\boldsymbol{\mu}_i - \boldsymbol{\mu}_j$ encodes the relative spatial displacement between the two groups; $\boldsymbol{\sigma}_i - \boldsymbol{\sigma}_j$ characterizes the discrepancy in local point spread; $1 - |\mathbf{n}_i^\top \mathbf{n}_j|$ quantifies the degree of normal inconsistency in a sign-invariant manner; and $|\kappa_i - \kappa_j|$ reflects the variation in local surface curvature. The resulting pairwise feature $\mathbf{e}_{ji}$ serves as the edge feature in the subsequent graph module.

Edge-aware Message Passing. The graph nodes correspond to the group features $\{\mathbf{x}_g\}_{g=1}^G$, and edges are constructed by connecting each node to its K nearest neighbors among the group centroids, forming a directed K-NN graph. We denote by $\mathcal{N}(i)$ the set of source nodes connected to node i. Let $\mathbf{x}_i^{(\ell)} \in \mathbb{R}^D$ denote the node feature of group i at layer ℓ, and let $\mathbf{e}_{ji}^{(\ell)}$ denote the edge feature encoding the pairwise relationship between groups j and i. At each layer, the edge representation is first refined by aggregating the features of both endpoint nodes alongside the current edge feature:

$$\tilde{\mathbf{e}}_{ji}^{(\ell)} = \mathrm{MLP}_e\left(\left[\mathbf{x}_i^{(\ell)} \,\|\, \mathbf{e}_{ji}^{(\ell)} \,\|\, \mathbf{x}_j^{(\ell)} \right] \right). \tag{2}$$

Concurrently, an attention coefficient for the directed edge $j \to i$ is computed from the destination-node query and the corresponding edge feature:

$$\alpha_{ji}^{(\ell)} = \mathrm{softmax}_{j \in \mathcal{N}(i)} \, \phi\left(\left[\mathrm{MLP}_q\left(\mathbf{x}_i^{(\ell)} \right) \,\|\, \mathrm{MLP}_r\left(\tilde{\mathbf{e}}_{ji}^{(\ell)} \right) \right] \right), \tag{3}$$

where $\mathrm{MLP}_q(\cdot)$ and $\mathrm{MLP}_r(\cdot)$ are learnable projections, $\phi(\cdot)$ is a learnable scoring function, and the softmax is normalized over all incoming neighbors $\mathcal{N}(i)$. The aggregated neighborhood signal at

node i is then obtained by summing attention-weighted messages from all incoming neighbors:

$$\mathbf{a}_i^{(\ell)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ji}^{(\ell)} \odot \mathrm{MLP}_v\left(\mathbf{x}_j^{(\ell)}\right), \tag{4}$$

where $\mathrm{MLP}_v(\cdot)$ is a learnable projection. The node feature is subsequently updated via a residual connection, $\mathbf{x}_i^{(\ell+1)} = \mathbf{x}_i^{(\ell)} + \mathrm{MLP}_n\Big(\Big[\,\mathbf{x}_i^{(\ell)} \,\|\, \mathbf{a}_i^{(\ell)}\,\Big]\Big)$, and the edge feature is similarly refined as $\mathbf{e}_{ji}^{(\ell+1)} = \mathbf{e}_{ji}^{(\ell)} + \tilde{\mathbf{e}}_{ji}^{(\ell)}$. Through this iterative process, the graph module progressively refines group-level representations by propagating relation-conditioned messages along directed edges and accumulating neighborhood information at each node, producing the final refined node features $\tilde{\mathbf{x}}_g$.

3.4 Cluster-Deviation Modeling

While **EGR** captures geometric relationships among neighboring regions, detecting anomalies also requires understanding whether a region is consistent within the broader distribution of structurally similar regions. To this end, we propose **Cluster-Deviation Modeling (CDM)**, as shown in Figure 3, which clusters regional features and quantifies each region's deviation from its cluster center. This allows the model to identify anomalies as regions that are relationally incompatible within a set of structurally similar normal structures, rather than relying solely on absolute feature values.

Node Feature Clustering. Let $\hat{\mathbf{x}}_g = \tilde{\mathbf{x}}_g / \|\tilde{\mathbf{x}}_g\|_2$ denote the ℓ_2-normalized node feature obtained after graph refinement. We perform clustering on $\{\hat{\mathbf{x}}_g\}_{g=1}^{G}$ into R clusters, where each cluster groups nodes with structurally similar feature representations. The cluster assignment of node g is denoted as $c(g) \in \{1, \ldots, R\}$, and the node set of the r-th cluster is defined as $\mathcal{C}_r = \{g \mid c(g) = r\}$. The center of each cluster is computed as the ℓ_2-normalized mean of its member node features:

$$\mathbf{z}_r = \mathrm{Norm}\left(\frac{1}{|\mathcal{C}_r|} \sum_{g \in \mathcal{C}_r} \hat{\mathbf{x}}_g\right), \tag{5}$$

where $\mathrm{Norm}(\cdot)$ denotes ℓ_2 normalization. The cluster center $\mathbf{z}_r$ serves as a reference representation for structurally similar regions within that cluster. Each node is then assigned the center of its corresponding cluster, $\bar{\mathbf{z}}_g = \mathbf{z}_r$ where $c(g) = r$, which is used in the subsequent deviation modeling. The same clustering procedure is applied to each test sample at inference time.

Distance-based Deviation Modeling. Given the assigned cluster center $\bar{\mathbf{z}}_g$, we quantify the deviation of each node from its cluster center via the cosine distance:

$$d_g = 1 - \hat{\mathbf{x}}_g^\top \bar{\mathbf{z}}_g. \tag{6}$$

The deviation d_g is then normalized within each cluster to produce a cluster-relative score, capturing how much each node deviates from the representative structure of its group. This relative deviation is more informative than absolute feature values, as anomalous regions are expected to exhibit greater inconsistency with respect to the surrounding normal structures. The scalar d_g is subsequently transformed into a sinusoidal positional embedding $\phi(d_g)$, which maps the scalar deviation into a high-dimensional representation to capture subtle deviation differences more expressively, and projected via a learnable linear map $\mathbf{W}_d$. The resulting deviation embedding is concatenated with the node feature to form the final node representation:

$$\mathbf{h}_g = [\,\tilde{\mathbf{x}}_g \,\|\, \mathbf{W}_d \phi(d_g)\,], \tag{7}$$

which is passed to a shared binary classifier to produce the anomaly score. To ensure that anomalous nodes produce deviation scores that are distinguishable from those of normal nodes within the same cluster, anomalous nodes are explicitly encouraged to deviate from their assigned cluster centers during training, as described in detail in Section 3.5.

3.5 Loss Function

Let $m_n \in \{0, 1\}$ denote the point-level anomaly label. We lift this supervision to the group level by defining $y_g = \max_{p_n \in \mathcal{P}^g} m_n$, such that a group is labeled anomalous if it contains at least one

Table 1: Quantitative comparison of mean AUROC (%) at both the object level and point level on Anomaly-ShapeNet and Real3D-AD. The best and second-best results are highlighted in **bold** and underline, respectively. Per-category results are provided in the supplementary material.

Method		Anomaly-ShapeNet		Real3D-AD	
		O-AUROC(↑)	P-AUROC(↑)	O-AUROC(↑)	P-AUROC(↑)
Category-Specific	BTF(Raw) (CVPR'23) [14]	49.3	55.0	63.5	57.1
	BTF(FPFH) [14]	52.8	62.8	60.3	73.3
	M3DM (CVPR'23) [30]	55.2	61.6	59.4	62.0
	PatchCore(FPFH) (CVPR'22) [28]	56.8	58.0	59.3	68.2
	PatchCore(PointMAE) [28]	56.2	57.7	59.4	62.0
	CPMF (PR'24) [5]	55.9	57.3	58.6	75.8
	IMRNet (CVPR'24) [17]	66.1	65.0	72.5	-
	Reg3D-AD (NeurIPS'23) [20]	57.2	66.8	70.4	70.5
	Group3AD (ACM MM'24) [44]	-	-	75.1	73.5
	R3D-AD (ECCV'24) [43]	74.9	-	73.4	-
	ISMP (AAAI'25) [18]	-	69.1	76.7	83.6
	PO3AD (CVPR'25) [33]	83.9	89.8	76.5	65.0
	PASDF (ICCV'25) [42]	90.0	89.7	80.2	74.5
	Reg2Inv (NeurIPS'25) [36]	86.1	88.2	78.0	87.8
Unified	MC3D-AD (IJCAI'25) [6]	84.2	75.9	78.2	76.8
	GRIM (Ours)	**97.4**	**94.5**	**87.2**	**91.3**

Table 2: Quantitative comparison of mean AUROC (%) under the cross-domain setting. In each → pair, the left denotes the training set and the right denotes the test set. Values in ( ) indicate in-domain performance, and Δ denotes the performance gap between in-domain and cross-domain results.

Method	Anomaly-ShapeNet → Real3D-AD				Real3D-AD → Anomaly-ShapeNet			
	O-AUROC(↑)	Δ(↓)	P-AUROC(↑)	Δ(↓)	O-AUROC(↑)	Δ(↓)	P-AUROC(↑)	Δ(↓)
MC3D-AD	55.4 (78.2)	22.8	38.3 (76.8)	38.5	78.3 (84.2)	5.9	48.8 (75.9)	27.1
GRIM (Ours)	**83.6** (87.2)	**3.6**	**89.6** (91.3)	**1.7**	**91.7** (97.4)	**5.7**	**78.9** (94.5)	**15.6**

anomalous point. The classification loss is defined in a cluster-balanced manner to prevent dominant clusters from biasing the training signal:

$$\mathcal{L}_{\mathrm{BCE}} = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \frac{1}{|\mathcal{C}_r|} \sum_{g \in \mathcal{C}_r} \mathrm{BCE}(\ell_g,\, y_g), \tag{8}$$

where $\mathcal{R}$ denotes the set of all clusters and $\mathrm{BCE}(\cdot, \cdot)$ denotes binary cross-entropy applied to logits.

To encourage anomalous nodes to deviate from their assigned cluster centers, we additionally employ a deviation loss. Letting $u_g = \hat{\mathbf{x}}_g^\top \bar{\mathbf{z}}_g$ denote the cosine similarity between node g and its cluster center, the deviation loss penalizes anomalous nodes that remain overly close to their center:

$$\mathcal{L}_{\mathrm{dev}} = \frac{1}{|\mathcal{G}_1|} \sum_{g \in \mathcal{G}_1} \max(u_g - \delta,\, 0), \tag{9}$$

where $\mathcal{G}_1 = \{g \mid y_g = 1\}$ denotes the set of anomalous nodes and δ is a predefined margin. The overall training objective is a weighted sum of the two losses, $\mathcal{L} = \lambda_{\mathrm{bce}}\, \mathcal{L}_{\mathrm{BCE}} + \lambda_{\mathrm{dev}}\, \mathcal{L}_{\mathrm{dev}}$, where λ_{bce} and λ_{dev} are scalar weighting coefficients.

4 Experiments

4.1 Experiment Settings

Datasets and Evaluation Metrics. We evaluate on two benchmarks: Anomaly-ShapeNet [17], a synthetic dataset with 1,600 samples across 40 categories (4 normal training samples each), and Real3D-AD [20], a real-world dataset with 12 categories (4 normal training samples each). Notably, Real3D-AD uses single-view test scans despite 360° training captures, posing a more challenging evaluation setting. Following prior work [20, 17], we report the Area Under the Receiver Operating



Figure 4: Qualitative comparison across various object categories, showing in-domain results (trained and tested on Anomaly-ShapeNet) and cross-domain results (trained on Anomaly-ShapeNet and tested on Real3D-AD). Red colored regions indicate detected anomalies.

Characteristic Curve (AUROC) at both the object level (O-AUROC) and point level (P-AUROC) to measure detection and localization performance, where higher values indicate better performance.

Baseline Methods. We compare our method against a range of state-of-the-art approaches: BTF [14], M3DM [30], PatchCore [28], CPMF [5], IMRNet [17], Reg3D-AD [20], Group3AD [44], R3D-AD [43], ISMP [18], PO3AD [33], PASDF [42], Reg2Inv [36], and MC3D-AD [6]. Among these, MC3D-AD is a unified model trained jointly across all categories, where diverse normal distributions across categories must be handled within a single model, making it a fundamentally more challenging setting than the category-specific paradigm. The remaining methods follow a category-specific paradigm, requiring a separate model to be trained for each object category. Results for all baselines are taken from their respective papers or reproduced using publicly available code.

4.2 Quantitative Results

In-Domain. Table 1 reports the mean O-AUROC and P-AUROC across all categories on Anomaly-ShapeNet and Real3D-AD. Despite being trained as a single unified model, our method consistently surpasses all category-specific baselines. On Anomaly-ShapeNet, we achieve gains of **7.4** in O-AUROC over PASDF [42] and **4.7** in P-AUROC over PO3AD [33], the previous best-performing methods in each metric. The same trend holds on Real3D-AD, where our method outperforms PASDF by **7.0** in O-AUROC and Reg2Inv [36] by **3.5** in P-AUROC. Furthermore, while MC3D-AD [6] retains the training efficiency of a unified model, it lags behind category-specific methods in detection performance, revealing a trade-off. Our method overcomes this limitation: we outperform MC3D-AD by **13.2** and **18.6** in O/P-AUROC on Anomaly-ShapeNet, and by **8.9** and **14.4** on Real3D-AD. These results demonstrate that training efficiency and detection accuracy need not be at odds, affirming the potential of unified frameworks for 3D anomaly detection.

Cross-Domain. Table 2 evaluates generalization capability under a cross-domain setting, where the training and test datasets are disjoint. We consider two transfer directions — training on Anomaly-ShapeNet and evaluating on Real3D-AD without any fine-tuning or domain-specific adaptation, and the reverse — and compare against the prior unified method MC3D-AD [6]. When trained on Anomaly-ShapeNet and evaluated on Real3D-AD, our method outperforms MC3D-AD by **28.2** in O-AUROC and **51.3** in P-AUROC. Remarkably, our model even surpasses all baselines in Table 1 that are trained directly on Real3D-AD. Furthermore, while MC3D-AD suffers a large performance drop in the cross-domain setting (Δ O/P-AUROC: 22.8 / 38.5), our method remains highly stable (Δ O/P-AUROC: **3.6** / **1.7**), demonstrating that our method effectively learns category-agnostic defect cues that generalize across domains, consistent with our core claim. The reverse transfer direction exhibits a similar trend, further corroborating this finding.

4.3 Qualitative Results

Figure 4 presents qualitative comparisons between our method and MC3D-AD [6] under both in-domain and cross-domain settings. MC3D-AD, as a reconstruction-based approach lacking explicit

defect supervision, exhibits two characteristic failure modes: incorrectly identifying rare yet normal structure variations as anomalies, and failing to detect anomalous regions that resemble normal surfaces. These limitations are further exacerbated in the cross-domain setting, where MC3D-AD tends to misclassify the majority of points as anomalous. In contrast, our method consistently produces precise anomaly localization in both settings, validating that our approach successfully learns category-agnostic defect representations that generalize across domain shifts. An extensive set of additional qualitative results is provided in the supplementary material.

4.4 Discussion

Effect of CDM on Center Distance. Figure 5 compares the average distance to the cluster center between all features and anomalous features alone, evaluated on the Anomaly-ShapeNet test set. Anomalous features exhibit a notably higher mean distance (0.73) than overall average (0.48), confirming that anomalous features tend to deviate further from their cluster centers under CDM. This deviation serves as a discriminative cue that distinguishes anomalous points from normal ones, validating the design rationale behind CDM.

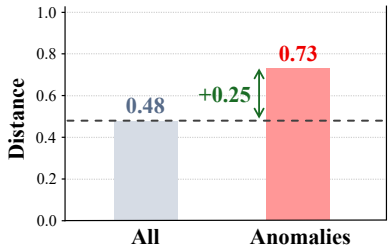


Figure 5: Average distances between features and cluster centers.

Analysis of Proposed Methods. Table 3 reports ablation results under both in-domain and cross-domain settings. Our baseline consists solely of a point backbone and a classifier. Adding Edge-aware Graph Refinement (EGR) yields consistent gains across both settings, with a **28.3** improvement in in-domain P-AUROC, demonstrating the effectiveness of modeling geometric relationships among local regions via graph-based representations. Adding Cluster-Deviation Modeling (CDM) further improves overall performance, with a pronounced gain of **22.1** in cross-domain P-AUROC, suggesting that exploiting relative deviation within clusters of structurally similar regions is more effective than relying solely on absolute feature representations for cross-domain generalization. When all components are combined, the substantial gains across all metrics indicate that the two modules operate synergistically, contributing to the performance.

Table 3: Ablation studies on the proposed components. O-AUROC (O-R) and P-AUROC (P-R) are reported on Anomaly-ShapeNet (A) and Real3D-AD (R).

EGR	CDM	In-Domain (A → A)		Cross-Domain (A → R)	
		O-R(↑)	P-R(↑)	O-R(↑)	P-R(↑)
		89.5	58.1	56.9	42.8
✓		94.3	86.4	59.4	53.0
	✓	96.6	79.8	62.3	64.9
✓	✓	**97.4**	**94.5**	**83.6**	**89.6**

Analysis of Anomaly Types. Table 4 compares different combinations of pseudo-anomaly types under in-domain and cross-domain settings. Notably, using any single anomaly type alone already yields competitive performance, with in-domain results remaining comparable to the full three-type setting and only a marginal cross-domain decline. Also, certain two-type combinations outperform the three-type configuration in-domain. Thus, increasing anomaly diversity does not produce monotonic improvements across all settings. These observations support the use of complementary perturbations as supervision for learning a transferable relational criterion, without requiring exhaustive simulation of real defect appearances. Here, the empirical sufficiency of the three primitives refers to their effectiveness as training supervision in the evaluated settings, rather than coverage of all real-world defect categories. Additional defect-type breakdowns and cross-type supervision experiments are presented in Section 6.3.

Table 4: Ablation studies on anomaly types.

Bulge	Sink	Hole	In-Domain (A → A)		Cross-Domain (A → R)	
			O-R(↑)	P-R(↑)	O-R(↑)	P-R(↑)
✓			96.9	94.1	77.8	88.0
	✓		98.5	95.9	81.2	81.8
		✓	95.9	93.9	78.4	86.8
✓	✓		96.9	94.1	81.8	89.0
✓		✓	98.6	96.0	80.3	87.9
	✓	✓	**99.2**	**98.1**	80.3	86.7
✓	✓	✓	97.4	94.5	**83.6**	**89.6**

5 Conclusion

We presented **GRIM**, a relational inconsistency modeling framework for 3D anomaly detection, built on the insight that defects manifest as violations of geometric consistency among neighboring structures rather than mere deviations from normality. Our two proposed modules, EGR and CDM, work synergistically to capture relational inconsistencies at both the structural and cluster levels, enabling the model to learn an explicit, category-agnostic defect criterion through pseudo-anomaly supervision. Experiments demonstrate improvements over prior methods across in- and cross-domain settings, validating relational inconsistency modeling as a promising direction for 3D anomaly detection.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00456589, RS-2026-25516683).

References

[1] David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pages 1027–1035, 2007.

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

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

[4] Luca Bonfiglioli, Marco Toschi, Davide Silvestri, Nicola Fioraio, and Daniele De Gregorio. The eyecandies dataset for unsupervised multimodal anomaly detection and localization. In *Proceedings of the Asian Conference on Computer Vision*, pages 3586–3602, 2022.

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

[6] 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 James Kwok, editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 837–845. International Joint Conferences on Artificial Intelligence Organization, 8 2025. Main Track.

[7] 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.

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

[9] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International conference on pattern recognition*, pages 475–489. Springer, 2021.

[10] Lei Fan, Junjie Huang, Donglin Di, Anyang Su, Tianyou Song, Maurice Pagnucco, and Yang Song. Salvaging the overlooked: Leveraging class-aware contrastive learning for multi-class anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21419–21428, 2025.

[11] Jia Guo, Shuai Lu, Weihang Zhang, Fang Chen, Huiqi Li, and Hongen Liao. Dinomaly: The less is more philosophy in multi-class unsupervised anomaly detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20405–20415, 2025.

[12] Haoyang He, Yuhu Bai, Jiangning Zhang, Qingdong He, Hongxu Chen, Zhenye Gan, Chengjie Wang, Xiangtai Li, Guanzhong Tian, and Lei Xie. Mambaad: Exploring state space models for multi-class unsupervised anomaly detection. *Advances in Neural Information Processing Systems*, 37:71162–71187, 2024.

[13] Dinh-Cuong Hoang, Phan Xuan Tan, Anh-Nhat Nguyen, Ta Huu Anh Duong, Tuan-Minh Huynh, Minh-Anh Nguyen, Duc-Manh Nguyen, Minh-Duc Cao, Duc-Huy Ngo, Minh-Quang Vu, et al. Vote3d-ad: Unsupervised point cloud anomaly localization via varied defect synthesis and differentiable vote-clustering. *Alexandria Engineering Journal*, 137:171–193, 2026.

[14] 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.

[15] Teng Hu, Jiangning Zhang, Ran Yi, Yuzhen Du, Xu Chen, Liang Liu, Yabiao Wang, and Chengjie Wang. Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 8526–8534, 2024.

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

[17] 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*, pages 22207–22216, 2024.

[18] 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.

[19] Hanzhe Liang, Jie Zhang, Tao Dai, Linlin Shen, Jinbao Wang, and Can Gao. Taming anomalies with down-up sampling networks: Group center preserving reconstruction for 3d anomaly detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 7133–7141, 2025.

[20] 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.

[21] 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.

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

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[24] Ruiying Lu, YuJie Wu, Long Tian, Dongsheng Wang, Bo Chen, Xiyang Liu, and Ruimin Hu. Hierarchical vector quantized transformer for multi-class unsupervised anomaly detection. *Advances in Neural Information Processing Systems*, 36:8487–8500, 2023.

[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] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[27] Nicolae-Cătălin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised predictive convolutional attentive block for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13576–13586, 2022.

[28] 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.

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

[30] 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.

[31] Shun Wei, Jielin Jiang, and Xiaolong Xu. Uninet: A contrastive learning-guided unified framework with feature selection for anomaly detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9994–10003, 2025.

[32] Xincheng Yao, Zixin Chen, Chao Gao, Guangtao Zhai, and Chongyang Zhang. Resad: A simple framework for class generalizable anomaly detection. *Advances in Neural Information Processing Systems*, 37:125287–125311, 2024.

[33] 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.

[34] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian conference on computer vision*, 2020.

[35] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. *Advances in Neural Information Processing Systems*, 35:4571–4584, 2022.

[36] 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.

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

[38] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112:107706, 2021.

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

[40] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. Destseg: Segmentation guided denoising student-teacher for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3914–3923, 2023.

[41] Ying Zhao. Omnial: A unified cnn framework for unsupervised anomaly localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3924–3933, 2023.

[42] 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.

[43] 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.

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

6 Appendix

Contents

- **6.1** Pseudo-Anomaly Generation Details
- **6.2** Implementation Details
- **6.3** Additional Discussion
- **6.4** Limitations and Future Works
- **6.5** Additional Quantitative Results
- **6.6** Additional Qualitative Results

6.1 Pseudo-Anomaly Generation Details

Algorithm 1 Pseudo-Anomaly Generation Algorithm

Input: Normal point cloud $\mathcal{P} = \{\mathbf{p}_n\}_{n=1}^{N}$
Output: Corrupted point cloud $\mathcal{P}^a$, anomaly mask $\mathcal{M} = \{m_n\}_{n=1}^{N}$
Initialize $\mathcal{P}^a \leftarrow \mathcal{P}$ and $\mathcal{M} \leftarrow \mathbf{0}$
Randomly sample a seed point from $\mathcal{P}$
Construct a local patch $\mathcal{Q}$ using K nearest neighbors of the seed point
Estimate the local surface normal $\mathbf{n}$ of $\mathcal{Q}$ by PCA
Randomly select anomaly type $\tau \in \{\texttt{bulge}, \texttt{sink}, \texttt{hole}\}$
if $\tau = \texttt{bulge}$ **then**
    Move points in $\mathcal{Q}$ outward along $\mathbf{n}$ with radial falloff
else if $\tau = \texttt{sink}$ **then**
    Move points in $\mathcal{Q}$ inward along $-\mathbf{n}$ with radial falloff
else if $\tau = \texttt{hole}$ **then**
    Move central points of $\mathcal{Q}$ toward the boundary
    Push central points slightly inward and expand boundary points mildly
end if
Update the modified patch in $\mathcal{P}^a$
Set $m_n = 1$ for modified points and $m_n = 0$ otherwise
return $\mathcal{P}^a, \mathcal{M}$

The overall procedure for pseudo-anomaly generation is summarized in Algorithm 1. For each normal point cloud, we synthesize pseudo-anomalies by locally perturbing a small neighborhood. A seed point is randomly selected from $\mathcal{P}$, and its nearest neighbors are used to form a local patch $\mathcal{Q}$. The local perturbation modifies approximately 1% of the points in each object. We estimate the local surface normal $\mathbf{n}$ of $\mathcal{Q}$ using PCA, where the eigenvector corresponding to the smallest eigenvalue is used as the normal direction. To make the perturbation direction consistent, the normal is oriented outward with respect to the global center of the point cloud.

The pseudo-anomaly generator supports three geometric perturbation primitives: `bulge`, `sink`, and `hole`. For `bulge` and `sink`, each point in the local patch is displaced along the estimated normal direction with a radial falloff:

$$\mathbf{q}_i^a = \mathbf{q}_i + \alpha s w_i \mathbf{n}, \tag{10}$$

where $\alpha = 1$ for `bulge` and $\alpha = -1$ for `sink`. Here, $s \sim \mathcal{U}(0.01, 0.03)$ denotes the deformation scale, and the radial falloff weight w_i is defined as

$$w_i = \left(1 - \frac{\|\mathbf{q}_i - \bar{\mathbf{q}}\|_2}{\max_j \|\mathbf{q}_j - \bar{\mathbf{q}}\|_2 + \epsilon}\right)^2, \tag{11}$$

where $\bar{\mathbf{q}}$ is the patch center. This falloff makes points near the patch center deform more strongly, while smoothly reducing the displacement toward the patch boundary.

For `hole`, the patch is divided into a central core region and a surrounding boundary ring. Core points are moved toward their nearest ring points and are additionally pushed inward along $-\mathbf{n}$, while ring points are mildly expanded along their radial directions. This produces a cavity-like local

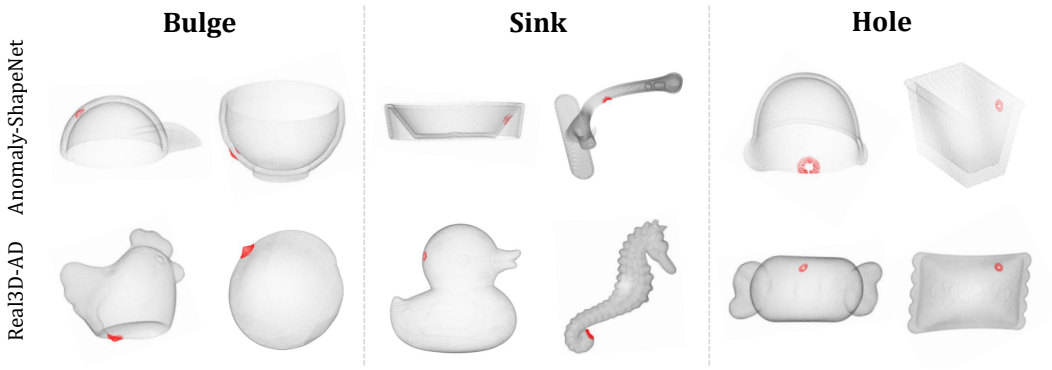


Figure 6: Visualization of pseudo-anomalies generated on Anomaly-ShapeNet and Real3D-AD. Gray points represent object point clouds, and red points denote the perturbed regions used as pseudo-anomaly masks.

deformation while preserving the overall locality of the perturbation. The anomaly mask $\mathcal{M}$ is set to one for synthetically modified points and zero otherwise. Representative examples of the generated pseudo-anomalies are illustrated in Figure 6.

After generating the pseudo-anomaly, we further apply a random rotation to the corrupted point cloud. The same rotation is applied to all points, while the point-level anomaly mask remains unchanged. This post-generation rotation encourages the model to learn pose-agnostic geometric inconsistency rather than relying on a fixed object orientation.

6.2 Implementation Details

Model setup. We adopt PointMAE [25], pretrained on ModelNet40 (8k), as the local feature extractor. The local feature extractor is initialized from the pretrained checkpoint and kept frozen during training. Each point cloud is partitioned into $G = 2048$ groups via FPS [26], with 128 neighboring points per group. For EGR, we construct a 32-NN graph over group centroids and employ 2 graph refinement layers with 4 attention heads and an attention dimension of 384. The edge feature has dimension 8, consisting of centroid difference (dimension 3), standard deviation difference (dimension 3), normal inconsistency (dimension 1), and curvature difference (dimension 1). We use summation aggregation with residual connections. For CDM, refined node features are ℓ_2-normalized and clustered into $R = 64$ centers using k-means++ [1] with 8 iterations. Node-to-center distances are normalized via quantile-based min-max normalization (0.1–0.9 quantiles), then encoded using a sinusoidal embedding with 128 dimensions (temperature 10000, scale 10), followed by a linear projection. The binary classifier is implemented as an MLP with hidden dimensions [256, 128] and dropout 0.1.

Anomaly scoring. For the i-th test point cloud $P_i = \{\mathbf{p}_{i,n}\}_{n=1}^{N_i}$, the binary classifier predicts a node-level anomaly score $a_{i,g}$ at each sampled group center $\mathbf{c}_{i,g}$, where $g \in \{1, \ldots, G\}$. We first propagate these scores to the original points by assigning each point the score of its nearest sampled node:

$$g_i^*(n) = \underset{g \in \{1,\ldots,G\}}{\arg\min} \|\mathbf{p}_{i,n} - \mathbf{c}_{i,g}\|_2, \qquad s_{i,n} = a_{i,g_i^*(n)}. \tag{12}$$

We then apply Gaussian smoothing to the propagated score vector $\mathbf{s}_i = [s_{i,1}, \ldots, s_{i,N_i}]$ and obtain the object-level anomaly score by taking the maximum of the smoothed point-wise anomaly map:

$$\widetilde{\mathbf{s}}_i = \mathrm{GaussianSmooth}_\sigma(\mathbf{s}_i), \qquad S_i^{\mathrm{obj}} = \max_{1 \le n \le N_i} \widetilde{s}_{i,n}. \tag{13}$$

No sample-wise min–max normalization is applied to the node-level or point-level anomaly scores before computing S_i^{obj}.

For evaluation, min–max normalization is applied only after the object-level scores have been computed, across evaluation samples within each category. Let $\mathcal{D}_c$ denote the index set of evaluation



Figure 7: t-SNE visualization of clustering results in CDM. Nodes of the same color belong to the same cluster.

samples belonging to category c. The normalized object-level score is

$$\widehat{S}_i^{\text{obj}} = \frac{S_i^{\text{obj}} - \min_{j \in \mathcal{D}_c} S_j^{\text{obj}}}{\max_{j \in \mathcal{D}_c} S_j^{\text{obj}} - \min_{j \in \mathcal{D}_c} S_j^{\text{obj}}}, \qquad i \in \mathcal{D}_c. \tag{14}$$

Training and evaluation. We train the model under the unified setting, where all categories are merged without using category labels. For each normal sample, we generate 4 pseudo-anomalous variants, with bulge, sink, and hole sampled with probabilities 0.4, 0.4, and 0.2, respectively. Random rotation is applied during training. We use AdamW [23] with a learning rate of 1×10^{-4}, weight decay 1×10^{-4}. Training is conducted for 100 epochs with a batch size of 1. The learning rate is decayed using StepLR. The loss weights are set to $\lambda_{\text{bce}} = 1.0$ and $\lambda_{\text{dev}} = 1.5$. Cross-domain evaluation is conducted by directly applying the trained model to the target domain without fine-tuning or domain-specific adaptation. All experiments are conducted with PyTorch 1.13.0 and CUDA 11.7 on a single NVIDIA GeForce RTX 3090 GPU, with a training time of approximately 12 hours.

6.3 Additional Discussion

Analysis of Clustering in CDM. Figure 7 presents a t-SNE visualization of the clustering results in **Cluster-Deviation Modeling (CDM)**, showing that structurally similar region features are well-separated into distinct clusters. Table 5 reports the effect of the number of clusters R on both in-domain and cross-domain performance. In the in-domain setting, performance improves generally as R increases, since finer clustering groups structurally similar regions more precisely, yielding more discriminative deviation scores. A similar trend is observed in the cross-domain setting up to $R = 64$.

Analysis of Deviation Weight in CDM. Table 6 reports the effect of the deviation loss weighting coefficient λ_{dev} on in-domain and cross-domain performance. As λ_{dev} increases, anomalous nodes are more strongly pushed away from their cluster centers, producing deviation scores that are more clearly distinguishable from those of normal nodes, which leads to general in-domain performance gains. In the cross-domain setting, the same tendency holds up to $\lambda_{\text{dev}} = 1.5$; beyond this point, however, an excessively large deviation penalty forces anomalous features to deviate in a manner that is overly tailored to the training distribution, which reduces generalizability under domain shift.

Table 5: Ablation studies on the number of clusters R on Anomaly-ShapeNet (A) and Real3D-AD (R).

R	In-Domain (A → A)		Cross-Domain (A → R)	
	O-AUROC(↑)	P-AUROC(↑)	O-AUROC(↑)	P-AUROC(↑)
16	96.1	89.9	63.8	73.3
32	96.8	92.5	68.4	79.0
64	**97.4**	**94.5**	**83.6**	**89.6**

Table 6: Ablation studies on the weighting coefficient λ_{dev}.

λ_{dev}	In-Domain (A → A)		Cross-Domain (A → R)	
	O-AUROC(↑)	P-AUROC(↑)	O-AUROC(↑)	P-AUROC(↑)
0.5	96.5	93.1	81.6	88.4
1.0	97.3	94.5	81.8	86.7
1.5	**97.4**	**94.5**	**83.6**	**89.6**

Table 7: Comparison of computational complexity.

Method		AIT ↓	FPS ↑	#Params ↓	FLOPs ↓
Category-Specific	PO3AD	**0.31s**	**3.23**	433.6M	**399**
	PASDF	0.92s	1.09	**15.9M**	624
	Reg2Inv	0.78s	1.28	85.3M	815
Unified	MC3D-AD	0.84s	1.19	34.2M	417
	GRIM (Ours)	**0.35s**	**2.86**	**34.0M**	**371**

Computational Complexity. Table 7 compares average inference time (AIT) and the number of parameters between our method and the category-specific models PO3AD [33], PASDF [42], and Reg2Inv [36], as well as the previous unified method MC3D-AD [6]. AIT is measured as the mean inference time over the 1,312 test samples of Anomaly-ShapeNet on a single NVIDIA RTX 3090 GPU. Our method achieves an AIT of 0.35s, which is more than twice as fast as MC3D-AD, demonstrating that our superior detection performance comes with no computational overhead — in fact, with a clear speed advantage. Furthermore, our model requires fewer parameters than MC3D-AD, highlighting its practical efficiency.

Table 8: Defect-type breakdown on Anomaly-ShapeNet.

Metric	Bulge	Concavity	Hole	Broken	Bending	Scratch
O-AUROC(↑)	96.1	95.9	97.0	96.9	96.6	97.7
P-AUROC(↑)	94.2	94.6	93.3	93.8	94.8	95.3

Generalization across Defect Types. We train the model on Anomaly-ShapeNet using bulge, sink, and hole pseudo-anomalies and evaluate it separately on the six test defect types: bulge, concavity, hole, broken, bending, and scratch. Bulge and hole directly correspond to training primitives, while concavity is geometrically similar to sink and is therefore not treated as an entirely unseen type. Broken, bending, and scratch are not explicitly synthesized by our generator. Table 8 reports the results. The model performs well on the defect types not explicitly synthesized during training; in particular, scratch achieves higher O-AUROC and P-AUROC than both bulge and hole. These results provide evidence of transfer across defect types beyond those directly instantiated by the training primitives.

Cross-Type Supervision. We further evaluate cross-type supervision on Real3D-AD by restricting pseudo-anomaly generation to the hole primitive. The Real3D-AD test set contains bulge and sink defects, but no hole defects. As shown in Table 9, hole-only supervision achieves 87.1 O-AUROC and 89.2 P-AUROC, compared with 87.2 and 91.3 when all three primitives are used. The corresponding reductions are 0.1 and 2.1 percentage points, respectively. This experiment supports transfer from the

Table 9: Cross-type supervision results on Real3D-AD.

Training pseudo-anomalies	O-AUROC ↑	P-AUROC ↑
Hole	87.1	89.2
Bulge, Sink, Hole	87.2	91.3

Table 10: Ablation of the deviation loss and deviation embedding in CDM.

Configuration	In-Domain ($A \to A$)		Cross-Domain ($A \to R$)	
	O-AUROC ↑	P-AUROC ↑	O-AUROC ↑	P-AUROC ↑
Full CDM	**97.4**	**94.5**	**83.6**	**89.6**
w/o $\mathcal{L}_{\text{dev}}$	96.6	92.1	79.3	84.1
w/o deviation emb.	94.9	88.7	68.5	66.8
w/o both	94.3	86.4	59.4	53.0

hole training primitive to the evaluated real defect types, without requiring an exact match between the training primitive and test defect type.

Contribution of Deviation Modeling in CDM. CDM uses the clustering procedure described in Section 6.2 to form feature-space peer groups. Its contribution lies in how these groups are used for anomaly discrimination, rather than in a new clustering algorithm. Specifically, $\mathcal{L}_{\text{dev}}$ encourages anomalous node features to deviate from their assigned cluster centers, while the deviation embedding $\mathbf{W}_d\phi(d_g)$ supplies the classifier with cluster-relative information alongside the EGR-refined node feature. Thus, the final anomaly score is predicted from the combined representation, not directly from the cluster deviation.

To examine these two roles, we keep the clustering procedure unchanged and remove the deviation loss and deviation embedding either individually or jointly. Table 10 reports results for in-domain evaluation on Anomaly-ShapeNet and cross-domain evaluation from Anomaly-ShapeNet to Real3D-AD. Removing either component reduces performance in both settings, with a larger cross-domain decrease when the deviation embedding is removed. Removing both components yields the lowest performance among the evaluated configurations. These results support the complementary contributions of supervised deviation shaping and deviation-feature injection within CDM.

Training Category-Specific Models in Unified Setting. In the category-specific setting, a separate model is trained and evaluated for each category. In contrast, unified training uses a single shared model across all categories without category labels or category-specific parameters. Although a unified model observes more training samples, it must accommodate heterogeneous normal distributions within a shared representation and anomaly criterion.

To examine the effect of this difference in training settings, we retrain PO3AD [33] and PASDF [42] under the unified protocol by merging the normal training samples from all categories. Table 11 compares the resulting performance with their category-specific counterparts on Real3D-AD and Anomaly-ShapeNet. Both methods exhibit lower O-AUROC and P-AUROC under unified training on both datasets, despite access to more training samples. These results indicate that pooling training data across categories does not automatically improve performance for the evaluated methods, highlighting the challenge of representing heterogeneous normal structures within a single shared model.

Additional Cross-Domain Comparison under Unified Training. We further evaluate PO3AD [33] and PASDF [42] under the unified cross-domain setting from Anomaly-ShapeNet to Real3D-AD. For each method, normal training samples from all Anomaly-ShapeNet categories are merged to train a single shared model without category labels. The source-trained model is then applied directly to Real3D-AD without fine-tuning or domain-specific adaptation.

Table 12 reports the results. Our method outperforms both additional unified baselines in object-level detection and point-level localization. This comparison further strengthens our claim of superior cross-domain performance.

Table 11: Comparison between category-specific and unified training on Real3D-AD and Anomaly-ShapeNet.(C) denotes the category-specific setting and (U) denotes the unified setting.

	Real3D-AD		Anomaly-ShapeNet	
Method	O-AUROC ↑	P-AUROC ↑	O-AUROC ↑	P-AUROC ↑
PO3AD (C)	**76.5**	**65.0**	**83.9**	**89.8**
PO3AD (U)	54.7	62.2	72.3	75.1
PASDF (C)	**80.2**	**74.5**	**90.0**	**89.7**
PASDF (U)	64.7	58.4	77.5	73.6

Table 12: Additional unified cross-domain comparison. (U) denotes the unified setting.

Method	O-AUROC ↑	P-AUROC ↑
PO3AD (U)	52.2	48.7
PASDF (U)	50.8	43.4
Ours	**83.6**	**89.6**

6.4 Limitations and Future Works

Overall. Our framework operates solely on point cloud data, which is an intentional design choice to focus on geometric relation-based defect detection. However, in practical industrial environments, RGB-D cameras naturally capture color and depth information alongside point clouds. Such complementary modalities can provide additional defect cues — such as surface color variations or subtle depth discontinuities — that are difficult to capture from geometry alone. Integrating multi-modal information into our relational inconsistency modeling framework therefore represents a promising direction for further improving detection performance, which we leave as future work.

Anomaly Sparsity in CDM. The cluster-relative cue in CDM relies on anomalies being sparse relative to normal structures. If an anomalous region is sufficiently large and coherent, its features may form a separate cluster, making their deviations from the resulting cluster center small. Consequently, the CDM signal may weaken under this condition. Although the classifier also receives EGR-refined node features, this additional input does not by itself establish robustness to large coherent anomalies. Reducing the number of clusters to form larger peer groups may be a possible direction for addressing this problem.

6.5 Additional Quantitative Results

Tables 13–14 and 15–16 report per-category O-AUROC and P-AUROC on Anomaly-ShapeNet and Real3D-AD, respectively. For cross-domain evaluation, Tables 17–18 cover the case of training on Anomaly-ShapeNet and testing on Real3D-AD, and Tables 19–20 cover the reverse direction. Table 21 additionally reports per-category object-level AUPR on Anomaly-ShapeNet.

6.6 Additional Qualitative Results

Figures 8–11 present additional in-domain qualitative results across diverse categories of Anomaly-ShapeNet, and Figures 12–13 present cross-domain results trained on Anomaly-ShapeNet and tested on Real3D-AD.

Table 13: Comparison of object-level AUROC (%) across all categories on Anomaly-ShapeNet.

	O-AUROC(↑)														
	Method	ashtray0	bag0	bottle0	bottle1	bottle3	bowl0	bowl1	bowl2	bowl3	bowl4	bowl5	bucket0	bucket1	cap0
Category-Specific	BTF(Raw)	57.8	41.0	59.7	51.0	56.8	56.4	26.4	52.5	38.5	66.4	41.7	61.7	32.1	66.8
	BTF(FPFH)	42.0	54.6	34.4	54.6	32.2	50.9	66.8	51.0	49.0	60.9	69.9	40.1	63.3	61.8
	M3DM	57.7	53.7	57.4	63.7	54.1	63.4	66.3	68.4	61.7	46.4	40.9	30.9	50.1	55.7
	PatchCore(FPFH)	58.7	57.1	60.4	66.7	57.2	50.4	63.9	61.5	53.7	49.4	55.8	46.9	55.1	58.0
	PatchCore(PointMAE)	59.1	60.1	51.3	60.1	65.0	52.3	62.9	45.8	57.9	50.1	59.3	59.3	56.1	58.9
	CPMF	35.3	64.3	52.0	48.2	40.5	78.3	63.9	62.5	65.8	68.3	68.5	48.2	60.1	60.1
	Reg3D-AD	59.7	70.6	48.6	69.5	52.5	67.1	52.5	49.0	34.8	66.3	59.3	61.0	75.2	69.3
	IMRNet	67.1	66.0	55.2	70.0	64.0	68.1	70.2	68.5	59.9	67.6	71.0	58.0	77.1	73.7
	R3D-AD	83.3	72.0	73.3	73.7	78.1	81.9	77.8	74.1	76.7	74.4	65.6	68.3	75.6	82.2
	PO3AD	**100.0**	83.3	90.0	93.3	92.6	92.2	82.9	83.3	88.1	**98.1**	84.9	85.3	78.7	87.7
	PASDF	**100.0**	99.5	**100.0**	**100.0**	**100.0**	**100.0**	94.8	**100.0**	**100.0**	93.3	91.2	**96.8**	77.5	85.2
	Reg2Inv	90.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	80.7	65.6	58.5	85.2	81.8	81.3	90.2	65.9
Unified	MC3D-AD	96.2	80.5	79.5	70.9	75.6	93.0	97.8	71.9	88.5	91.1	75.4	89.8	78.4	79.3
	GRIM (Ours)	**100.0**	97.1	**100.0**	99.3	94.3	95.9	**100.0**	93.3	97.8	96.7	**98.9**	95.9	**99.4**	**98.9**

	Method	cap3	cap4	cap5	cup0	cup1	eraser0	headset0	headset1	helmet0	helmet1	helmet2	helmet3	jar
Category-Specific	BTF(Raw)	52.7	46.8	37.3	40.3	52.1	52.5	37.8	51.5	55.3	34.9	60.2	52.6	42.0
	BTF(FPFH)	52.2	52.0	58.6	58.6	61.0	71.9	52.0	49.0	57.1	71.9	54.2	44.4	58.6
	M3DM	42.3	77.7	63.9	53.9	55.6	62.7	57.7	61.7	52.6	42.7	62.3	37.4	56.4
	PatchCore(FPFH)	45.3	75.7	79.0	60.0	58.6	65.7	58.3	63.7	54.6	48.4	42.5	40.4	49.4
	PatchCore(PointMAE)	47.6	72.7	53.8	61.0	55.6	67.7	59.1	62.7	55.6	55.2	44.7	42.4	48.3
	CPMF	55.1	55.3	69.7	49.7	49.9	68.9	64.3	45.8	55.5	58.9	46.2	52.0	61.0
	Reg3D-AD	72.5	64.3	46.7	51.0	53.8	34.3	53.7	61.0	60.0	38.1	61.4	36.7	59.2
	IMRNet	77.5	65.2	65.2	64.3	75.7	54.8	72.0	67.6	59.7	60.0	64.1	57.3	78.0
	R3D-AD	73.0	68.1	67.0	77.6	75.7	89.0	73.8	79.5	75.7	72.0	63.3	70.7	83.8
	PO3AD	85.9	79.2	67.0	87.1	83.3	99.5	80.8	92.3	76.2	96.1	86.9	75.4	86.6
	PASDF	64.9	64.6	85.3	97.1	85.7	95.2	**100.0**	79.5	81.2	93.8	76.5	84.6	**100.0**
	Reg2Inv	86.3	68.1	90.2	73.3	93.3	**100.0**	**100.0**	84.3	81.7	98.6	87.5	87.6	**100.0**
Unified	MC3D-AD	70.1	83.5	76.1	74.3	95.2	77.6	86.2	88.6	67.2	**100.0**	60.9	**97.9**	97.1
	GRIM (Ours)	**99.6**	**98.6**	**98.2**	**98.6**	**100.0**	84.8	98.2	**96.2**	**97.7**	99.0	**99.4**	94.8	**100.0**

	Method	phone	shelf0	tap0	tap1	vase0	vase1	vase2	vase3	vase4	vase5	vase7	vase8	vase9	Mean
Category-Specific	BTF(Raw)	56.3	16.4	52.5	57.3	53.1	54.9	41.0	71.7	42.5	58.5	44.8	42.4	56.4	49.3
	BTF(FPFH)	67.1	60.9	56.0	54.6	34.2	21.9	54.6	69.9	51.0	40.9	51.8	66.8	26.8	52.8
	M3DM	35.7	56.4	75.4	73.9	57.4	42.7	73.7	43.9	47.6	43.9	65.7	66.3	66.3	55.2
	PatchCore(FPFH)	38.8	49.4	75.3	76.6	60.4	42.3	72.1	44.9	50.6	44.9	69.3	66.2	66.0	56.8
	PatchCore(PointMAE)	48.8	52.3	45.8	53.8	51.3	55.2	74.1	46.0	51.6	46.0	65.0	66.3	62.9	56.2
	CPMF	50.9	68.5	35.9	69.7	58.2	34.5	58.2	58.2	51.4	58.2	39.7	52.9	60.9	55.9
	Reg3D-AD	41.4	68.8	67.6	64.1	48.6	70.2	60.5	65.0	50.0	65.0	46.2	62.0	59.4	57.2
	IMRNet	75.5	60.3	67.6	69.6	55.2	75.7	61.4	70.0	52.4	70.0	63.5	63.0	59.4	66.1
	R3D-AD	76.2	69.6	73.6	90.0	73.3	72.9	75.2	74.2	63.0	74.2	77.1	72.1	71.8	74.9
	PO3AD	77.6	57.3	74.5	68.1	85.8	74.2	95.2	82.1	67.5	85.2	96.6	73.9	83.0	83.9
	PASDF	**100.0**	71.3	88.2	79.3	**100.0**	92.9	**100.0**	80.6	91.2	**100.0**	**100.0**	92.4	83.6	90.0
	Reg2Inv	**100.0**	57.7	94.8	80.4	99.6	60.5	**100.0**	84.5	81.8	**100.0**	64.3	81.8	87.3	86.1
Unified	MC3D-AD	91.9	84.1	94.5	**97.0**	79.5	85.7	92.9	76.1	87.6	76.1	93.8	67.0	73.6	84.2
	GRIM (Ours)	95.2	**95.4**	**99.4**	91.1	**100.0**	**97.6**	95.2	**97.6**	**99.4**	**100.0**	**100.0**	**95.8**	**96.7**	**97.4**

Table 14: Comparison of point-level AUROC (%) across all categories on Anomaly-ShapeNet.

	P-AUROC(↑)													
	Method	ashtray0	bag0	bottle0	bottle1	bottle3	bowl0	bowl1	bowl2	bowl3	bowl4	bowl5	bucket0	bucket1
Category-Specific	BTF(Raw)	51.2	43.0	55.1	49.1	72.0	52.4	46.4	42.6	68.5	56.3	51.7	61.7	68.6
	BTF(FPFH)	62.4	74.6	64.1	54.9	62.2	71.0	76.8	51.8	59.0	67.9	69.9	40.1	63.3
	M3DM	57.7	63.7	66.3	63.7	53.2	65.8	66.3	69.4	65.7	62.4	48.9	69.8	69.9
	PatchCore(FPFH)	59.7	57.4	65.4	68.7	51.2	52.4	53.1	62.5	32.7	72.0	35.8	45.9	57.1
	PatchCore(PointMAE)	49.5	67.4	55.3	60.6	65.3	52.7	52.4	51.5	58.1	50.1	56.2	58.6	57.4
	CPMF	61.5	65.5	52.1	57.1	43.5	74.5	48.8	63.5	64.1	68.3	68.4	48.6	60.1
	Reg3D-AD	69.8	71.5	88.6	69.6	52.5	77.5	61.5	59.3	65.4	80.0	69.1	61.9	75.2
	IMRNet	67.1	66.8	55.6	70.2	64.1	78.1	70.5	68.4	59.9	57.6	71.5	58.5	77.4
	PO3AD	**96.2**	94.9	91.2	84.4	88.0	97.8	91.4	**91.8**	93.5	96.7	94.1	75.5	89.9
	PASDF	91.9	95.8	95.1	92.6	94.8	96.3	90.0	81.6	93.9	86.5	90.9	87.5	82.4
	Reg2Inv	78.5	**99.1**	**99.5**	84.9	81.7	**98.3**	82.8	82.2	76.1	78.8	82.4	61.0	85.5
Unified	MC3D-AD	80.1	81.5	89.5	88.3	90.1	82.7	53.4	60.7	78.3	65.4	55.2	79.0	89.5
	GRIM (Ours)	89.9	95.0	98.1	**94.7**	**97.4**	96.5	**92.5**	90.0	**95.6**	**97.1**	**96.5**	**89.9**	**95.1**

	Method	cap0	cap3	cap4	cap5	cup0	cup1	eraser0	headset0	headset1	helmet0	helmet1	helmet2	helmet3
Category-Specific	BTF(Raw)	52.4	68.7	46.9	37.3	63.2	56.1	63.7	57.8	47.5	50.4	44.9	60.5	70.0
	BTF(FPFH)	73.0	65.8	52.4	58.6	79.0	61.9	71.9	62.0	59.1	57.5	74.9	64.3	72.4
	M3DM	53.1	60.5	71.8	65.5	71.5	55.6	71.0	58.1	58.5	59.9	42.7	62.3	65.5
	PatchCore(FPFH)	47.2	65.3	59.5	79.5	65.5	59.6	81.0	58.3	46.4	54.8	48.9	45.5	73.7
	PatchCore(PointMAE)	54.4	48.8	72.5	54.5	51.0	85.6	37.8	57.5	42.3	58.0	56.2	65.1	61.5
	CPMF	60.1	55.1	55.3	55.1	49.7	50.9	68.9	69.9	45.8	55.5	54.2	51.5	52.0
	Reg3D-AD	63.2	71.8	81.5	46.7	68.5	69.8	75.5	58.0	62.6	60.0	62.4	82.5	62.0
	IMRNet	71.5	70.6	75.3	74.2	64.3	68.8	54.8	70.5	47.6	59.8	60.4	64.4	66.3
	PO3AD	95.7	94.8	94.0	86.4	90.9	93.2	97.4	82.3	90.7	87.8	**94.8**	93.2	84.6
	PASDF	94.8	86.1	89.4	92.0	94.8	88.4	94.5	86.3	89.1	81.6	64.6	80.9	**95.8**
	Reg2Inv	86.1	94.5	86.4	**97.0**	79.8	88.1	**98.0**	**94.6**	**97.0**	**92.5**	90.6	89.1	95.6
Unified	MC3D-AD	86.7	92.2	88.0	88.5	82.9	71.5	85.6	66.6	70.9	74.4	58.2	81.7	60.2
	GRIM (Ours)	**97.8**	**97.6**	**94.3**	94.3	**97.5**	**97.0**	90.4	92.9	93.8	90.7	88.1	**95.1**	88.7

	Method	jar	phone	shelf0	tap0	tap1	vase0	vase1	vase2	vase3	vase4	vase5	vase7	vase8	vase9	Mean
Category-Specific	BTF(Raw)	42.3	58.3	46.4	52.7	56.4	61.8	54.9	40.3	60.2	61.3	58.5	57.8	55.0	56.4	55.0
	BTF(FPFH)	42.7	67.5	61.9	56.8	59.6	64.2	61.9	64.6	69.9	71.0	42.9	54.0	66.2	56.8	62.8
	M3DM	54.1	35.8	55.4	65.4	71.2	60.8	60.2	73.7	65.8	65.5	64.2	51.7	55.1	66.3	61.6
	PatchCore(FPFH)	47.8	48.8	61.3	73.3	76.8	65.5	45.3	72.1	43.0	50.5	44.7	69.3	57.5	66.3	58.0
	PatchCore(PointMAE)	48.7	88.6	54.3	85.8	54.1	67.7	55.1	74.2	46.5	52.3	57.2	65.1	36.4	42.3	57.7
	CPMF	61.1	54.5	78.3	45.8	65.7	45.8	48.6	58.2	58.2	51.4	65.1	50.4	52.9	54.5	57.3
	Reg3D-AD	59.9	59.9	68.8	58.9	74.1	54.8	60.2	40.5	51.1	75.5	62.4	88.1	81.1	69.4	66.8
	IMRNet	76.5	74.2	60.5	68.1	69.9	53.5	68.5	61.4	40.1	52.4	68.2	59.3	63.5	69.1	65.0
	PO3AD	87.1	81.0	66.3	78.3	69.2	95.5	88.2	97.8	88.4	90.2	93.7	98.2	95.0	95.2	89.8
	PASDF	95.9	95.1	86.5	88.4	**90.2**	94.4	79.7	95.6	86.8	89.9	91.5	95.9	90.9	86.3	89.7
	Reg2Inv	**98.2**	**99.2**	63.2	**91.8**	86.9	**98.0**	70.5	**99.7**	84.4	92.7	87.9	86.3	93.4	**97.1**	88.2
Unified	MC3D-AD	84.8	85.4	65.8	52.4	53.7	87.0	72.6	82.2	79.7	78.6	64.7	61.6	88.5	77.6	75.9
	GRIM (Ours)	97.5	97.3	**93.5**	90.1	88.1	97.9	**96.0**	95.8	**97.4**	**97.2**	**94.0**	**98.9**	**97.4**	94.1	**94.5**

Table 15: Comparison of object-level AUROC (%) across all categories on Real3D-AD.

	O-AUROC(↑)													
	Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Mean
Category-Specific	BTF(Raw)	73.0	64.7	53.9	78.9	70.7	69.1	60.2	68.6	59.6	39.6	53.0	70.3	63.5
	BTF(FPFH)	52.0	56.0	63.0	43.2	54.5	78.4	54.9	64.8	77.9	75.4	57.5	46.2	60.3
	M3DM	43.4	54.1	55.2	68.3	60.2	43.3	54.0	64.4	49.5	69.4	55.1	45.0	59.4
	PatchCore(FPFH)	**88.2**	59.0	54.1	83.7	57.4	54.6	67.5	37.0	50.5	58.9	44.1	56.5	59.3
	PatchCore(PointMAE)	72.6	49.8	66.3	82.7	78.3	48.9	63.0	37.4	53.9	50.1	51.9	58.5	59.4
	CPMF	70.1	55.1	55.2	50.4	52.3	58.2	55.8	58.9	72.9	65.3	70.0	39.0	58.6
	IMRNet	76.2	71.1	75.5	78.0	90.5	51.7	88.0	67.4	60.4	66.5	67.4	77.4	72.5
	Reg3D-AD	71.6	69.7	68.5	85.2	90.0	58.4	91.5	41.7	76.2	58.3	50.6	82.7	70.4
	Group3AD	74.4	72.8	84.7	78.6	93.2	67.9	97.6	53.9	84.1	58.5	56.2	79.6	75.1
	R3D-AD	77.2	69.6	71.3	71.4	68.5	**90.9**	69.2	66.5	72.0	84.0	70.1	70.3	73.4
	ISMP	85.8	73.1	85.2	71.4	94.8	71.2	94.5	46.8	72.9	62.3	66.0	84.2	76.7
	PO3AD	80.4	65.4	78.5	68.6	80.1	82.0	85.9	69.3	75.6	80.0	75.8	77.1	76.5
	PASDF	62.8	**95.9**	78.8	73.9	89.4	65.8	**98.9**	63.4	**100.0**	85.0	61.7	**86.6**	80.2
	Reg2Inv	81.8	75.8	**100.0**	**94.4**	**100.0**	75.0	67.2	73.5	53.2	69.2	84.1	62.6	78.0
Unified	MC3D-AD	85.0	74.9	83.0	71.5	95.5	83.1	86.5	56.0	71.6	80.3	76.6	73.8	78.2
	GRIM (Ours)	80.5	79.9	93.8	87.2	99.1	81.5	91.1	**80.0**	82.6	**96.5**	**91.2**	82.6	**87.2**

Table 16: Comparison of point-level AUROC (%) across all categories on Real3D-AD.

	P-AUROC(↑)													
	Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Mean
Category-Specific	BTF(Raw)	56.4	64.7	73.5	60.9	56.3	60.1	51.4	59.7	52.0	48.9	39.2	62.3	57.1
	BTF(FPFH)	73.8	70.8	86.4	73.5	88.2	87.5	70.9	89.1	51.2	57.1	50.1	81.5	73.3
	M3DM	54.7	60.2	67.9	67.8	60.8	66.7	60.6	67.4	56.0	73.8	53.2	68.2	62.0
	PatchCore(FPFH)	56.2	75.4	78.0	42.9	82.8	26.4	82.9	91.0	73.9	73.9	60.6	74.7	68.2
	PatchCore(PointMAE)	56.9	60.9	62.7	72.9	71.8	52.8	71.7	44.4	63.3	70.9	58.0	58.0	62.0
	CPMF	61.8	83.6	73.4	55.9	75.3	71.9	**98.8**	44.9	**96.2**	72.5	80.0	**95.9**	75.8
	Reg3D-AD	63.1	71.8	72.4	67.6	83.5	50.3	82.6	54.5	81.7	81.1	61.7	75.9	70.5
	Group3AD	63.6	74.5	73.8	75.9	86.2	63.1	83.6	56.4	82.7	79.8	62.5	80.3	73.5
	ISMP	75.3	83.6	90.7	79.8	92.6	87.6	88.6	85.7	81.3	83.9	64.1	89.5	83.6
	PO3AD	71.5	57.4	76.3	56.3	61.4	64.4	87.7	53.0	58.5	54.2	65.9	72.9	65.0
	PASDF	77.7	80.2	54.6	76.8	69.9	78.2	83.7	65.4	88.7	64.8	70.3	83.8	74.5
	Reg2Inv	**92.3**	**94.4**	**96.9**	**91.0**	**97.9**	93.7	84.6	90.7	64.5	90.6	84.0	73.7	87.8
Unified	MC3D-AD	62.8	81.9	91.0	64.0	94.2	82.2	93.2	45.8	65.9	77.8	69.0	93.4	76.8
	GRIM (Ours)	75.9	87.6	94.4	90.7	97.6	**94.1**	97.3	**92.5**	87.0	**96.2**	**87.4**	94.5	**91.3**

Table 17: Comparison of object-level AUROC (%) across all categories on Real3D-AD under the cross-domain setting. Models are trained on Anomaly-ShapeNet and tested on Real3D-AD.

O-AUROC(↑)													
Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Mean
MC3D-AD	42.8	48.2	58.8	50.5	67.0	63.3	49.5	57.1	57.8	59.4	65.6	45.4	55.4
GRIM (Ours)	**78.0**	**70.2**	**76.4**	**81.2**	**96.2**	**85.9**	**81.6**	**81.0**	**92.8**	**97.1**	**88.0**	**74.6**	**83.6**

Table 18: Comparison of point-level AUROC (%) across all categories on Real3D-AD under the cross-domain setting. Models are trained on Anomaly-ShapeNet and tested on Real3D-AD.

P-AUROC(↑)													
Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Mean
MC3D-AD	38.5	25.0	32.1	44.1	37.4	36.5	35.9	41.2	36.3	53.8	50.1	29.1	38.3
GRIM (Ours)	**76.3**	**87.0**	**92.0**	**84.6**	**93.9**	**90.1**	**95.2**	**92.2**	**86.1**	**97.3**	**89.6**	**91.3**	**89.6**

Table 19: Comparison of object-level AUROC (%) across all categories on Anomaly-ShapeNet under the cross-domain setting. Models are trained on Real3D-AD and tested on Anomaly-ShapeNet.

O-AUROC(↑)																				
Method	ashtray0	bag0	bottle0	bottle1	bottle3	bowl0	bowl1	bowl2	bowl3	bowl4	bowl5	bucket0	bucket1	cap0	cap3	cap4	cap5	cup0	cup1	eraser0
MC3D-AD	62.9	68.1	77.6	80.0	78.1	**97.8**	73.7	75.2	74.1	78.5	76.1	74.6	75.2	61.9	89.8	87.0	80.4	74.8	86.2	**80.0**
GRIM (Ours)	**96.2**	**79.0**	**85.7**	**82.8**	**99.7**	97.0	**91.1**	**100.0**	**93.7**	**91.5**	**95.4**	**94.0**	**85.1**	**87.0**	**100.0**	**100.0**	**95.4**	**91.0**	**91.4**	**80.0**

Method	headset0	headset1	helmet0	helmet1	helmet2	helmet3	jar	phone	shelf0	tap0	tap1	vase0	vase1	vase2	vase3	vase4	vase5	vase7	vase8	vase9	Mean
MC3D-AD	86.2	81.0	77.7	79.5	76.8	**92.1**	**93.3**	85.2	65.5	67.9	71.5	86.3	70.0	90.5	80.3	85.2	81.0	78.1	70.3	63.3	78.3
GRIM (Ours)	**95.6**	**95.7**	**84.9**	**96.2**	**94.5**	84.8	88.6	**92.4**	**82.0**	**98.5**	**97.0**	**97.9**	**78.6**	**94.8**	**85.2**	**99.4**	**92.4**	**80.0**	**92.7**	**98.8**	**91.7**

Table 20: Comparison of point-level AUROC (%) across all categories on Anomaly-ShapeNet under the cross-domain setting. Models are trained on Real3D-AD and tested on Anomaly-ShapeNet.

P-AUROC(↑)													
Method	ashtray0	bag0	bottle0	bottle1	bottle3	bowl0	bowl1	bowl2	bowl3	bowl4	bowl5	bucket0	bucket1
MC3D-AD	48.6	51.3	56.2	62.7	48.5	49.5	43.4	48.6	43.1	50.3	43.6	55.3	47.3
GRIM (Ours)	**82.1**	**85.6**	**86.5**	**78.6**	**84.9**	**75.6**	**61.7**	**84.6**	**77.0**	**79.3**	**81.6**	**77.1**	**80.8**

Method	cap0	cap3	cap4	cap5	cup0	cup1	eraser0	headset0	headset1	helmet0	helmet1	helmet2	helmet3
MC3D-AD	46.8	54.2	48.7	48.3	54.0	36.5	56.5	47.1	55.0	49.6	37.0	53.7	64.8
GRIM (Ours)	**87.6**	**86.7**	**81.6**	**86.6**	**87.5**	**81.0**	**84.9**	**78.8**	**79.6**	**86.0**	**75.3**	**74.8**	**74.6**

Method	jar	phone	shelf0	tap0	tap1	vase0	vase1	vase2	vase3	vase4	vase5	vase7	vase8	vase9	Mean
MC3D-AD	36.3	52.2	51.8	44.1	46.5	46.8	**65.9**	44.0	46.8	48.1	44.2	41.4	47.0	35.3	48.8
GRIM (Ours)	**77.5**	**85.8**	**82.0**	**76.2**	**75.6**	**80.9**	65.7	**64.3**	**81.2**	**78.7**	**72.1**	**80.6**	**59.0**	**74.6**	**78.9**

Table 21: Comparison of object-level AUPR (%) across all categories on Anomaly-ShapeNet.

		O-AUPR(↑)												
	Method	ashtray0	bag0	bottle0	bottle1	bottle3	bowl0	bowl1	bowl2	bowl3	bowl4	bowl5	bucket0	bucket1
Category-Specific	BTF(Raw)	57.8	45.8	46.6	57.3	54.3	58.8	46.4	57.6	65.4	60.1	61.5	65.2	62.0
	BTF(FPFH)	65.1	55.1	64.4	62.5	60.2	57.6	64.8	51.5	49.9	63.2	69.9	48.3	64.8
	M3DM	63.2	64.2	76.3	67.4	45.1	52.5	51.5	63.0	63.5	57.1	60.1	60.9	50.7
	PatchCore(FPFH)	44.5	60.8	61.5	67.7	57.9	54.8	54.5	61.1	62.0	57.5	54.1	60.4	56.5
	PatchCore(PointMAE)	67.9	60.1	54.5	64.5	65.1	56.2	61.1	45.6	55.6	60.1	58.5	54.1	64.2
	CPMF	45.3	65.5	58.8	59.2	50.5	77.5	62.1	60.1	41.8	68.3	68.5	66.2	50.1
	Reg3D-AD	58.8	60.8	63.2	69.5	47.4	49.4	51.5	49.5	44.1	62.4	55.5	63.2	71.4
	IMRNet	61.2	66.5	55.8	70.2	64.8	48.1	50.4	68.1	61.4	63.0	65.2	57.8	73.2
	PO3AD	99.9	80.9	92.7	95.9	**96.2**	94.6	90.5	88.8	92.7	**98.5**	90.4	92.3	88.2
Unified	**GRIM (Ours)**	**100.0**	**91.6**	**95.6**	**100.0**	92.7	**95.1**	**100.0**	**98.4**	**95.3**	**98.5**	**100.0**	**98.0**	**100.0**

	Method	cap0	cap3	cap4	cap5	cup0	cup1	eraser0	headset0	headset1	helmet0	helmet1	helmet2	helmet3
Category-Specific	BTF(Raw)	65.9	61.2	51.5	65.3	60.1	70.1	42.5	37.9	51.5	55.9	38.8	61.5	52.6
	BTF(FPFH)	61.8	57.9	54.5	59.3	58.5	65.1	71.9	53.1	52.3	56.8	72.1	58.8	56.4
	M3DM	56.4	65.2	47.7	64.2	57.0	75.2	62.5	63.2	62.3	52.8	62.7	63.6	45.8
	PatchCore(FPFH)	58.5	45.7	65.5	72.5	60.4	58.6	58.4	70.1	60.1	52.5	63.0	47.5	49.4
	PatchCore(PointMAE)	56.1	58.3	72.1	54.2	64.2	71.0	80.1	51.5	42.3	63.3	57.1	49.6	61.1
	CPMF	60.1	54.1	64.5	69.7	64.7	60.9	54.4	60.2	61.9	33.3	50.1	47.7	64.5
	Reg3D-AD	69.3	71.1	62.3	77.0	53.1	63.8	42.4	53.8	61.7	60.0	38.1	61.8	46.8
	IMRNet	71.1	70.2	65.8	50.2	45.5	62.7	59.9	70.1	65.6	69.7	61.5	60.2	57.5
	PO3AD	84.1	90.6	87.6	80.1	87.9	87.0	**99.5**	76.5	91.4	86.4	96.1	93.4	84.9
Unified	**GRIM (Ours)**	**100.0**	**99.7**	**96.3**	**96.0**	**100.0**	**100.0**	97.9	**98.4**	**99.5**	**97.8**	98.0	**99.8**	**96.3**

	Method	jar	phone	shelf0	tap0	tap1	vase0	vase1	vase2	vase3	vase4	vase5	vase7	vase8	vase9	Mean
Category-Specific	BTF(Raw)	42.8	61.3	62.4	53.5	59.4	56.2	44.1	41.3	71.7	42.8	61.5	54.7	41.6	48.2	54.9
	BTF(FPFH)	47.9	66.2	61.1	61.0	57.5	64.1	65.5	56.9	65.2	58.7	47.2	59.2	62.4	63.8	59.8
	M3DM	55.5	46.4	66.5	72.2	63.8	78.8	65.2	61.5	55.1	52.6	63.3	64.8	46.3	65.1	60.3
	PatchCore(FPFH)	49.9	33.2	50.4	71.2	68.4	64.5	62.3	80.1	48.1	77.7	51.5	62.1	51.5	66.0	58.8
	PatchCore(PointMAE)	46.3	65.2	54.3	71.2	54.2	54.8	57.2	71.1	45.5	58.6	58.5	65.2	65.5	63.4	59.5
	CPMF	61.8	65.5	68.1	63.9	69.7	63.2	64.5	63.2	58.8	65.5	51.8	43.2	67.3	61.8	59.7
	Reg3D-AD	60.1	61.4	67.5	67.6	59.9	61.5	46.8	64.1	65.1	50.5	58.8	45.5	62.9	57.4	58.4
	IMRNet	76.0	55.2	62.5	40.1	79.6	57.3	72.5	65.5	70.8	52.8	65.4	60.1	63.9	46.2	62.1
	PO3AD	91.5	80.3	68.0	85.6	70.9	75.3	78.9	**96.3**	90.2	82.4	87.9	97.1	83.3	90.4	88.1
Unified	**GRIM (Ours)**	**100.0**	**88.2**	**93.0**	**100.0**	**81.7**	**96.5**	**91.8**	95.9	**98.4**	**100.0**	**100.0**	**99.5**	**98.8**	**93.6**	**97.1**

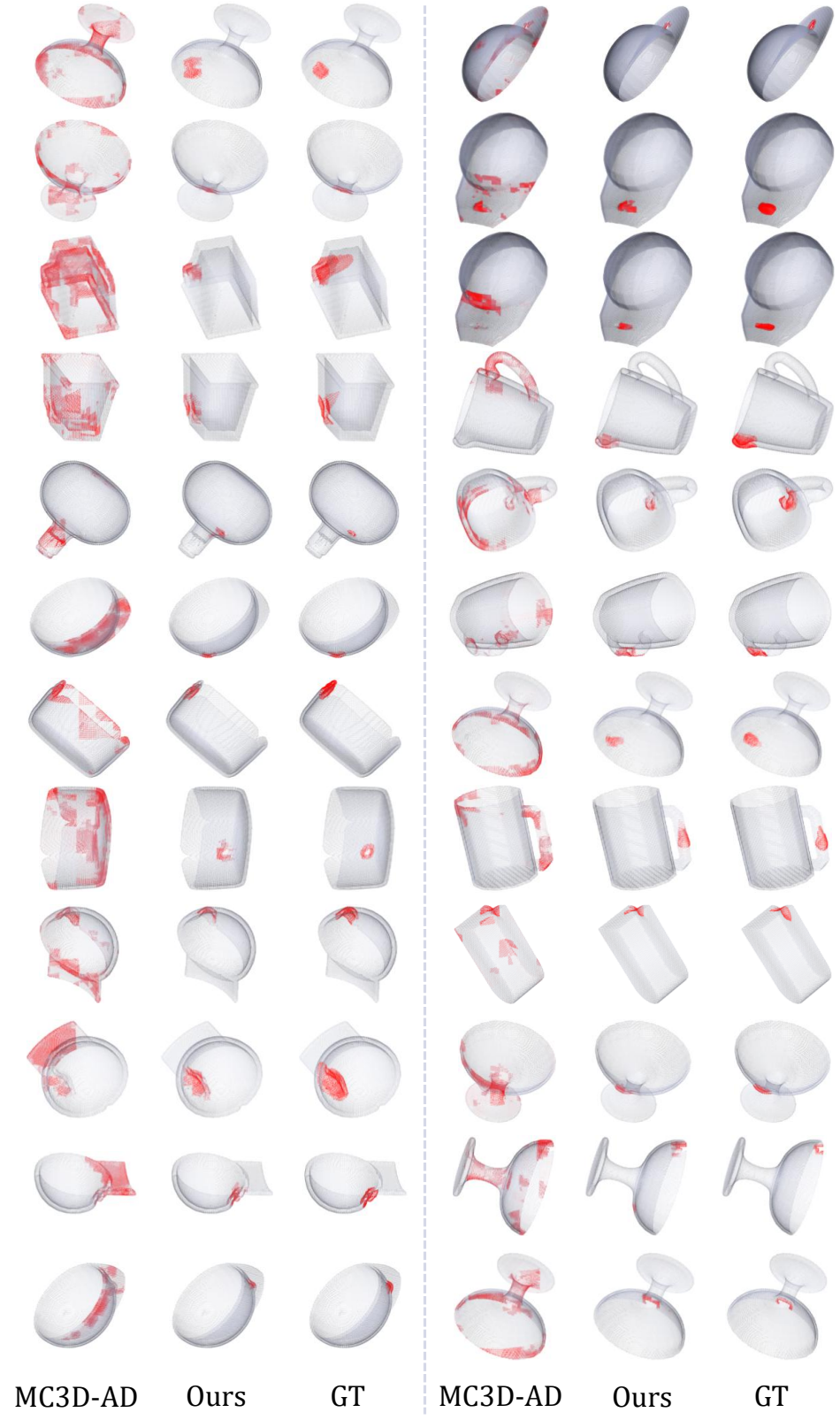


Figure 8: Additional in-domain qualitative comparison with MC3D-AD on Anomaly-ShapeNet.

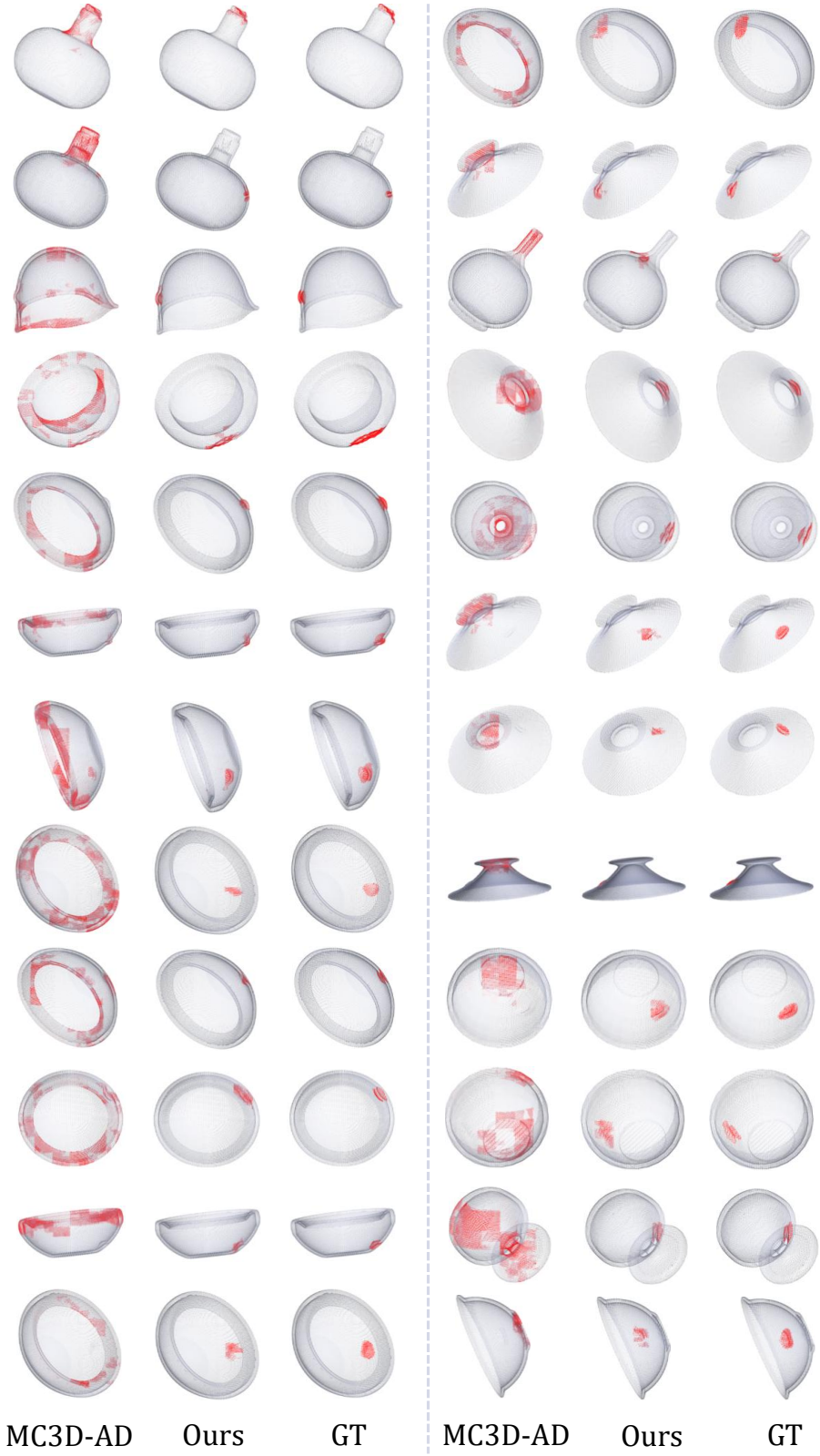


Figure 9: Additional in-domain qualitative comparison with MC3D-AD on Anomaly-ShapeNet.

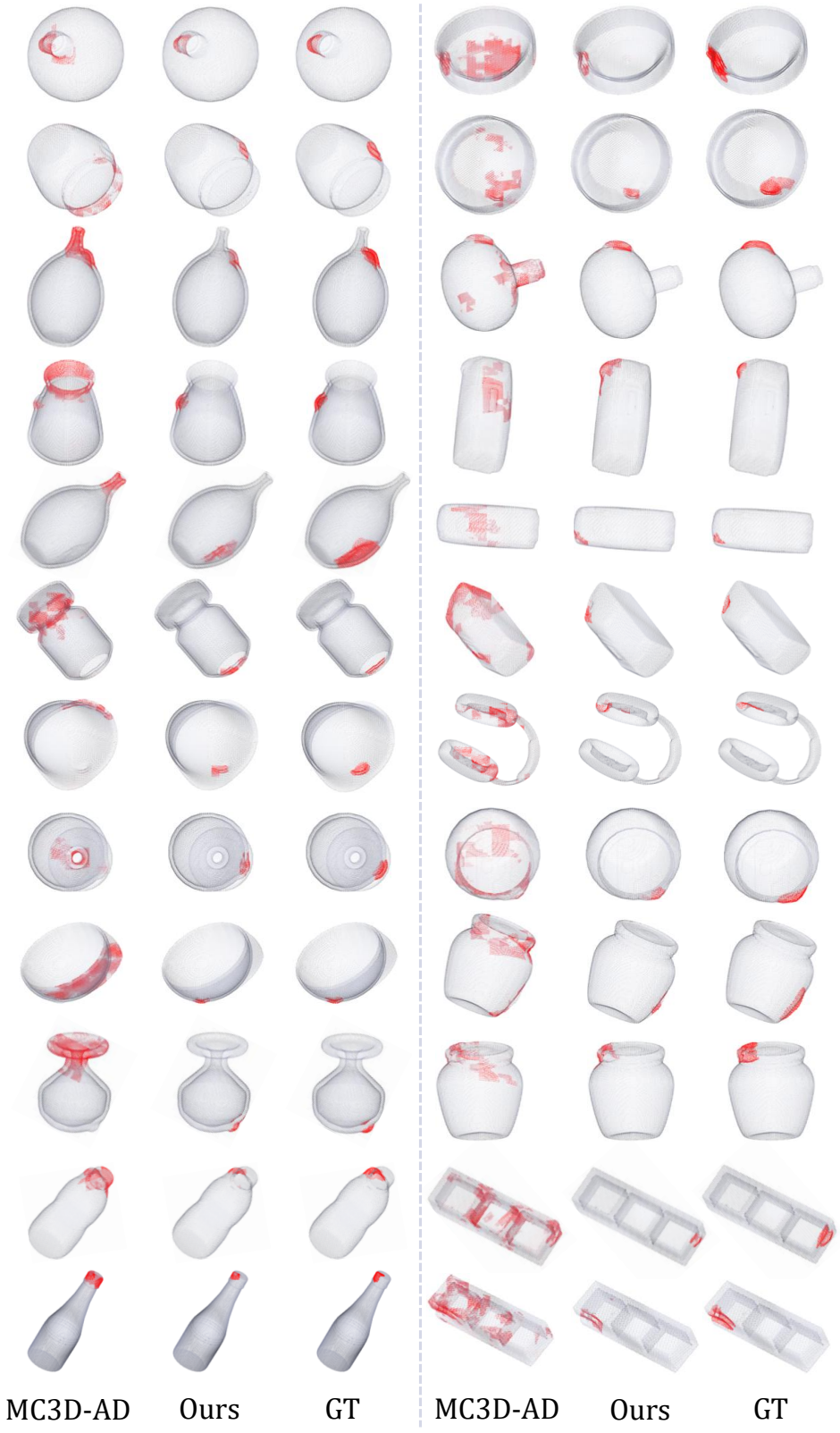


Figure 10: Additional in-domain qualitative comparison with MC3D-AD on Anomaly-ShapeNet.

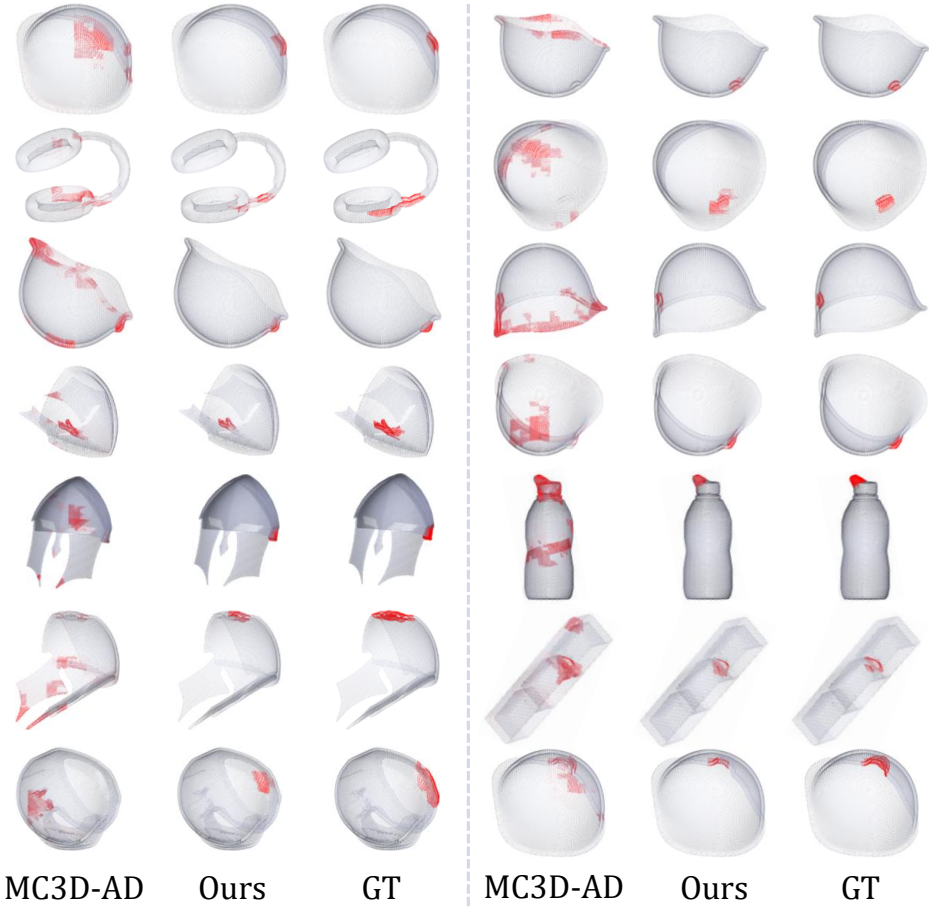


Figure 11: Additional in-domain qualitative comparison with MC3D-AD on Anomaly-ShapeNet.

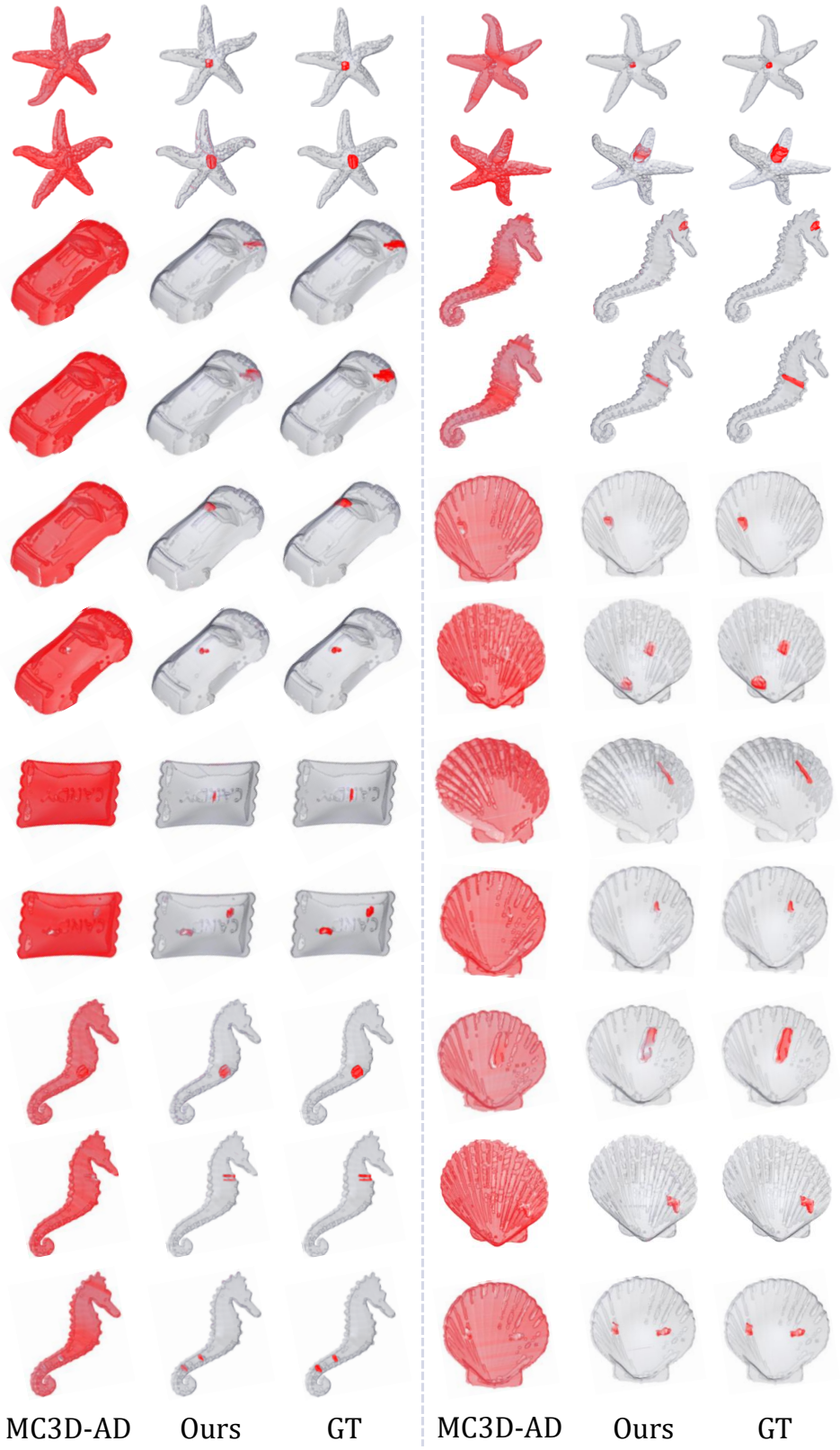


Figure 12: Additional cross-domain qualitative comparison with MC3D-AD. Models are trained on Anomaly-ShapeNet and tested on Real3D-AD.

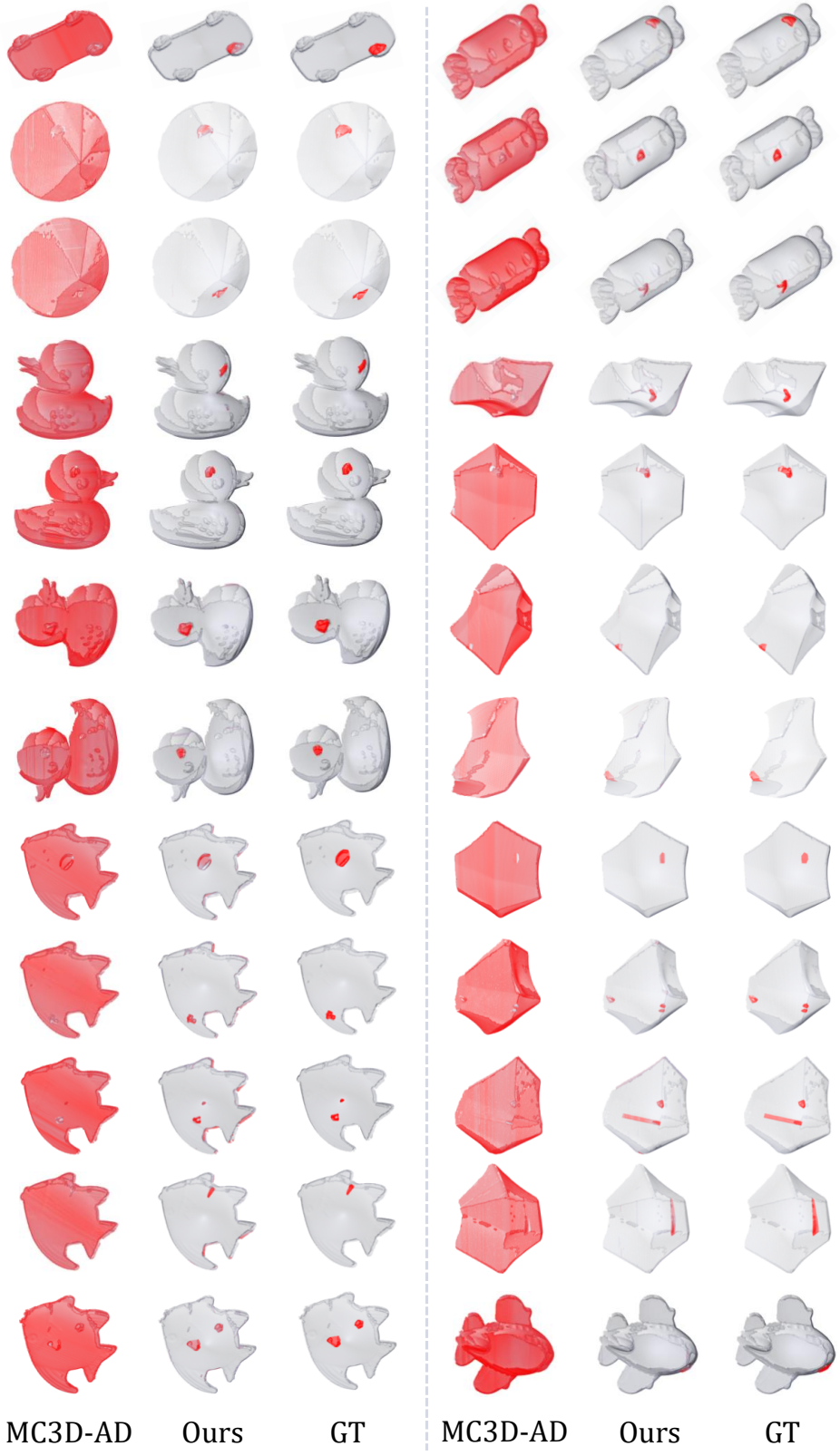


Figure 13: Additional cross-domain qualitative comparison with MC3D-AD. Models are trained on Anomaly-ShapeNet and tested on Real3D-AD.

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 claims made in the abstract and introduction accurately reflect the contributions of the paper, consistent with experimental results presented.

   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: The paper presents limitations that include a discussion of potential issues.

   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: The paper does not include theoretical results requiring formal proofs.

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: All experimental details necessary to reproduce the main results are described in the paper. Code will be made publicly available upon publication.

   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: [No]

Justification: Code is not released at submission time, but will be made publicly available with sufficient instructions upon publication.

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: All training and evaluation details, including data splits, hyperparameters, and optimizer settings, are described in the paper.

   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: Error bars are not reported in this paper. However, our experiments produce highly deterministic results — repeated runs yield nearly identical outcomes, indicating negligible variance. Error bars can be provided upon request if needed.

   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: The paper provides sufficient information to reproduce the experiments, including GPU type, training time, and other relevant compute details.

   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: The research conforms with the NeurIPS Code of Ethics in all respects.

   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: [No]

    Justification: This paper focuses on technical contributions and does not include a specific discussion of broader positive or negative societal impacts.

    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 paper does not release models or datasets that pose a high risk of 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: All existing assets used in this paper, including datasets and code, are properly credited with appropriate citations and licenses acknowledged.

    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: New assets introduced in this paper will be documented and released publicly upon publication.

    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: The paper does not involve crowdsourcing or 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: The paper does not involve human subjects research and therefore does not require IRB approval.

    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: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of the core methodology in this research.

    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.