The Good, the Bad, and the Template: Contrastive Anomaly Detection in 3D

Alexander Tarvo[1], Colin Acton[1], Yusen Wan[1], and Xu Chen[1]

University of Washington, Seattle WA 98109, USA,
{alexta,actonc,yusenwan,chx}@uw.edu

Abstract. Anomaly detection and localization (ADL) in point clouds is a rapidly expanding field of 3D computer vision, owing to its importance in robotic manufacturing and automated quality control. Recent ADL methods extract representations of 3D geometries from a test object and compare them to representations of anomaly-free objects. Despite rapid progress, modern ADL techniques still struggle to meet accuracy expectations in safety-critical fields such as aerospace. The main challenges are learning representations that are highly robust for anomaly detection and developing algorithms that accurately and unambiguously compare these representations. We overcame the first challenge by formulating ADL as a semi-supervised contrastive learning problem and developing a deep representation extractor optimized for anomaly detection. For the second challenge, we compare test representations with anomaly-free representations of multiple reference objects, precisely aligned in a common 3D reference frame. Our method establishes a new state of the art on Real3D-AD and Anomaly-Shapenet datasets, achieving a mean area under the ROC curve of 91.2% and 94.9%, respectively.

1 Introduction

Anomaly detection in 3D point clouds—identifying geometric deviations from a nominal shape—is essential across a multitude of applications, from obstacle detection in autonomous systems to quality assurance in precision manufacturing. In safety-critical domains such as aerospace, undetected defects can have severe consequences, demanding highly accurate anomaly detection methods. Unlike 2D imagery, where anomaly detection has reached considerable maturity, the 3D domain presents distinct challenges: point clouds are sparse, unordered, and lack the regular grid structure characteristic of 2D images. Geometric defects often manifest as subtle surface variations—a shallow scratch, a slight deformation, etc.—that are difficult to distinguish from normal geometry. These difficulties are exacerbated by severe class imbalance: for instance, anomalous regions in the Real3D-AD dataset occupy only 1–5% of object surfaces [12].

Two main research paradigms have emerged for 3D anomaly detection. *Reconstruction-based methods* train a generative model to re-create the anomaly-free "template" point cloud from a "test" input. Regions where the reconstruction deviates significantly from the test object are flagged as anomalies. *Patch-based*

methods divide template objects into local patches, extract per-patch representations, and store them in a *memory bank*. Significant deviations between representations extracted from the test object and the contents of a memory bank indicate the presence of an anomaly.

Despite their architectural differences, both paradigms face the same problem of reliably comparing a test point cloud or its learned representations against a reference. The unstructured, noisy nature of point clouds makes this comparison difficult: reconstruction-based methods are limited by reconstruction fidelity, while patch-based methods suffer from imprecise representations and ambiguous spatial matching.

Specifically, patch-based methods exhibit three key limitations. *First*, many anomalies correspond to subtle geometric deviations, yet existing approaches rely on general-purpose descriptors—such as Fast Point Feature Histograms (FPFH) [15] or pre-trained neural backbones—that are not tailored to detecting such deviations. *Second*, methods that employ a global memory bank suffer from *spatial ambiguity*; they may compare patches from different locations within an object. This ambiguity hinders the detection of anomalies arising from misplaced but otherwise valid geometric features (e.g., a hole drilled at an incorrect location). *Third*, while 3D anomaly detection is often formulated as an unsupervised problem to enable learning from limited data, real-world anomalies exhibit structured geometric patterns—e.g., bumps, scratches, and holes—that introduce strong inductive biases. These biases, inherent to 3D perception in robotics and manufacturing, have not been sufficiently exploited by existing methods.

We address these shortcomings by formulating 3D anomaly detection as a semi-supervised contrastive learning problem. We leverage sparse anomaly annotations available in standard 3D ADL benchmarks—typically reserved for evaluation—to learn expressive, spatially *aware* geometric representations. This enables reliable detection of subtle geometric defects while avoiding spatial ambiguity, leading to what we shall refer to as COSSAD-3D—a COntrastive Semi-Supervised Anomaly Detector in 3D. COSSAD-3D advances the state of the art in three major aspects:

- **Semi-supervised contrastive feature extraction.** We formulate anomaly detection as a semi-supervised contrastive learning problem and develop a deep feature extractor that is highly optimized for detecting geometric anomalies. Our extractor minimizes the distance between representations of geometrically similar patches and maximizes the distance between anomalous representations.
- **Multi-objective learning.** We define the training objective as a weighted sum of loss functions: a *strong* annotation-driven objective that enables precise detection of subtle anomalies and a *weak* geometry-driven component that improves robustness to out-of-distribution geometries. We establish an analytical convergence criterion for the combined objective.
- **Spatially aware patch comparison.** We design an anomaly detection algorithm that does not suffer from spatial ambiguity and is robust to the representations' noise. We define a multitude of small local memory banks, each bank containing features extracted from a specific location of multiple

reference objects. We compare a patch originating from a "test" object only against the contents of the corresponding local memory bank.

COSSAD-3D establishes a new accuracy baseline on both the Anomaly-ShapeNet and Real3D-AD datasets. The average area under the ROC curve (AUROC) on the Real3D-AD dataset is 91.2% for COSSAD-3D, compared to 82.9% for the strongest baseline, PointCore. AUROC on the Anomaly-ShapeNet dataset is 94.9%, compared to the 86.0% reported by the Simple3D algorithm. The COSSAD-3D code is accessible at `https://github.com/alextarvo/cossad`.

2 Related Work

Contrastive learning extracts representations by contrasting tuples of positive (similar) and negative (dissimilar) instances, pulling positives closer together in the embedding space while pushing negatives apart. The specific form of this objective is governed by the choice of loss function. Triplet loss [16] contrasts a pair of positive instances against a single negative. InfoNCE [13] extends this by contrasting a positive pair against multiple negatives, maximizing mutual information between different views of the data.

These earlier approaches relied on external selection ("mining") of the most expressive tuples. SimCLR [2], an unsupervised technique, reduces the need for mining by employing strong data augmentation and large batch sizes. MoCo [7] then reduced the memory footprint of SimCLR by introducing a momentum encoder and a memory queue.

Supervised contrastive losses [9] enable learning on labeled data. CircleLoss [17] and Multi Similarity (MS) loss [20], for example, make tuple mining unnecessary. MS loss provides additional flexibility by allowing custom groupings of positive/negative tuples.

Semi-supervised and weakly supervised contrastive methods address settings where full label information is unavailable. These approaches either infer pseudo-labels during training or exploit weaker supervisory signals already present in the data. In [21], unsupervised contrastive loss is combined with supervised cross-entropy defined on learned pseudo-labels to jointly train on labeled and unlabeled data. In [25], a "strong" self-supervised InfoNCE objective is augmented with a "weak" supervised contrastive loss operating on graph-derived pseudo-labels. Cl-InfoNCE [19] imputes the "weak" labeling signal from the dataset metadata.

Rather than relying on pseudo-labels—which require careful calibration and are sensitive to distribution shift—we decompose the contrastive objective into a "strong" component that leverages anomaly annotations to precisely align positive and negative instances in the tuple, and a "weak" component that exploits geometric correspondence. We analytically establish criteria on tuple composition required for training convergence. To the best of our knowledge, this is the first application of semi-supervised contrastive learning to 3D anomaly detection.

3D anomaly detection and localization (ADL) can be broadly categorized into reconstruction-based and patch-based (also known as feature- or representation-based) methods.

Let PC denote a point cloud. *Reconstruction-based methods*, such as R3D-AD [26] or IMRNet [10], train a model to reproduce "good" PCs and use the trained model to reconstruct an anomalous PC. The model, trained on good shapes, will not properly reconstruct an anomalous shape, resulting in high reconstruction loss and hence recognition of the defects. More specifically, MC3D-AD [4] trains a generative transformer on synthetic anomalies and compares the resulting 3D shape to a reference. PO3DAD [22] combines a reconstruction-based backbone with a supervised regression head that predicts displacement vectors between a reference and a test shape.

Patch-based models learn representations (i.e., features) of PCs. They assume that in the feature space, normal parts cluster together and anomalies appear as outliers. Among representative models, 3D-ST [1] relies on a student-teacher model to learn representative patch features and uses the L2 loss between the outputs of a student and a teacher as the anomaly score. M3DM [28] combines 2D and 3D patches to learn rich representations. Building on the PatchCore [14] 2D algorithm, patch-based models extract features from both good and anomalous PCs, store them in a memory bank, and then match them using the min-max selection algorithm. BTF [8] and Simple3D [5] use high-quality, handcrafted multi-scale features. Liu et al. [12] found that the accuracy of [14], [8], and [28] significantly fluctuates between object types. Their Reg3D-AD, the first registration-based ADL algorithm, improved accuracy substantially. PointCore [24] also relies on registration; it uses point interpolation to compare a test object to a single reference PC. ICMP [11] combines 3D and 2D representations obtained by projecting a PC along the z-axis and then applies PatchCore to a pair of memory banks representing the feature and spatial spaces. Group3D [27] assigns representations to a pre-defined number of clusters. It increases separation ("contrast" in the authors' terms) between clusters by explicitly optimizing inter- and intra-cluster variance using gradient descent.

3 Proposed Spatial-Aware Defect Detection

The proposed method takes *object classes* (e.g., a bowl, a bucket, or a microphone) as input. Objects from the same class will contain rigid bodies with standard shapes and dimensions, and are represented by their PCs. Each PC X is an unordered set of n points $X = \{x_i\}_{i=1}^{n}, x_i \in \mathcal{R}^3$. We register all PCs to a global reference frame and split each PC into patches. A *patch* $P_i^j \subset X^j$ is a spherical region of a PC X^j with a center $c_i \in \mathcal{R}^3$ and a radius R.

Let X^t denote a "template" anomaly-free PC, X^g another "good" anomaly-free PC, and X^b an anomalous "bad" PC with *anomaly mask* $\tilde{X}^b \subset X^b$—the set of all anomalous points in X^b. Then P_i^b is a patch with a center $c_i \in \tilde{X}^b$, so that P_i^b is extracted from the anomalous region $\tilde{X}^b$ of a "bad" PC X^b. P_i^b must be geometrically different from the patch P_i^t extracted from *exactly the same* location c_i of a "template" PC X^t. However, the patch P_i^g extracted from the "good" PC X^g must be similar to P_i^t.

We observe that patches P are unordered sets of 3D points that are not directly comparable. Instead, we compare their representation vectors $\mathbf{z} \in \mathcal{R}^d$ produced by a feature extractor—a neural network f_θ with parameters θ, such that $\mathbf{z} = f_\theta(P)$.

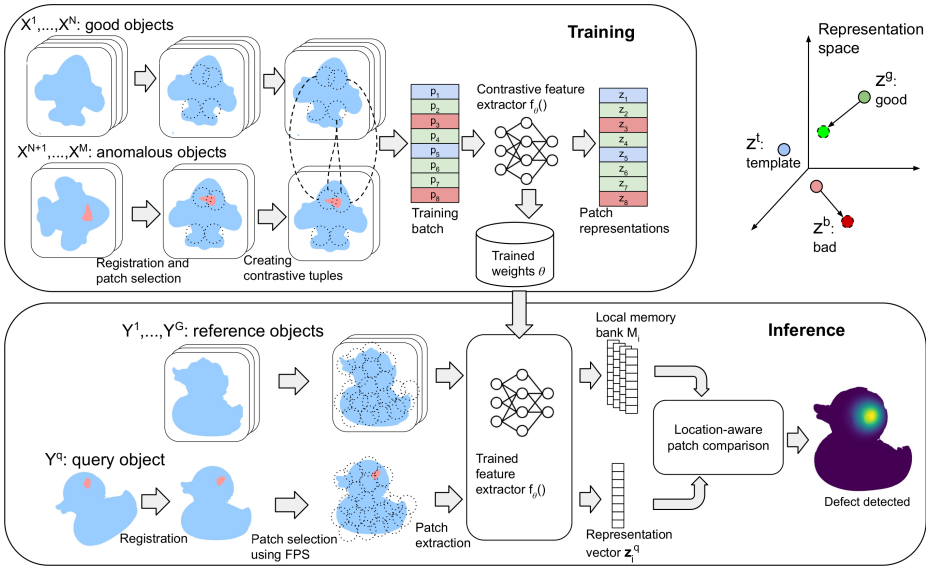

Fig. 1: Overview of COSSAD-3D ADL system. During the training, we extract patches from anomaly-free "good" and anomalous "bad" PCs. We arrange patches into contrastive tuples and train a deep feature extractor using a contrastive loss. During inference, a trained extractor fetches patch representations from the reference objects and the object under test. An anomaly detection algorithm compares patches in a spatially-aware manner to localize anomalies.

The proposed COSSAD-3D has two distinct modes of operation: training and inference (see Fig. 1). The input to the training stage is known object classes and their PCs. For each class, $X^1, \cdots, X^N$ denote anomaly-free, "good" PCs. $X^{N+1}, \cdots, X^M$ represent anomalous, "bad" PCs, with anomaly masks $\tilde{X}^{N+1}, \cdots, \tilde{X}^M$ as defined above.

We train a feature extractor that minimizes the similarity between the representation $\mathbf{z}_i^t = f_\theta(P_i^t)$ of a template patch P_i^t, $t \in (1, \cdots, N)$, and the representation of a corresponding anomalous patch $\mathbf{z}_i^b = f_\theta(P_i^b)$, $b \in (N+1, \cdots, M)$. At the same time, the extractor maximizes the similarity between $\mathbf{z}_i^t$ and the representation of an anomaly-free, "good" patch $\mathbf{z}_i^g = f_\theta(P_i^g)$, $g \in (1, \cdots, N), g \neq t$. More formally, we require $S(\mathbf{z}_i^t, \mathbf{z}_i^g) >> S(\mathbf{z}_i^t, \mathbf{z}_i^b)$, where $S(\cdot, \cdot)$ is a similarity function such as cosine similarity.

The input to the inference stage is the set of anomaly-free "reference" PCs $Y^1, \cdots, Y^G$ and a "query" PC of an object under test Y^q. COSSAD-3D registers $Y^1, \cdots Y^G, Y^q$ in the global reference frame, splits them into patches, and extracts representations using a trained extractor $f_\theta(\cdot)$. Reference representations are

stored in memory banks, having a separate memory bank $\mathcal{M}_i$ for each patch location c_i. For each "query" patch P_i^q, COSSAD-3D computes an anomaly score δ_i by comparing $f_\theta(P_i^q)$ with the contents of $\mathcal{M}_i$. Finally, it aggregates anomaly scores for all patches into a single anomaly score Δ for the query object Y^q.

3.1 Semi-supervised contrastive feature extractor

Our feature extractor is a deep learning network trained with contrastive loss. The input here is a batch $\mathcal{B} = \{p_1, \cdots, p_m\}$ of patches; $p_a \in \mathbb{R}^{n_{points} \times 3}$. The output is the patch representations $\mathcal{Z} = \{\mathbf{z}_1, \cdots, \mathbf{z}_m\}$. $\mathbf{z}_a = f_\theta(p_a); \mathbf{z}_a \in \mathbb{R}^d$.

Let $\mathbf{h} \in \mathcal{R}^D$ be a patch descriptor vector produced by an underlying neural backbone such as PointNet++, RIConv++, or a similar point cloud backbone. We replace the backbone's task-specific head with the contrastive head $\mathbf{z} = g_{\theta_s}(\mathbf{h})$, which accepts $\mathbf{h}$ and produces the L2-normalized patch representation $\mathbf{z} \in \mathcal{R}^d$:

$$\begin{aligned} \mathbf{h}_1 &= \mathrm{ReLU}(\mathrm{BN}_D(\mathrm{Dropout}_{p=0.4}(\mathbf{h}))), && \mathbf{h}_1 \in \mathbb{R}^{B\times D}, \\ \mathbf{h}_2 &= \mathrm{ReLU}(\mathrm{BN}_{256}(\mathrm{Linear}_{D\to 256}(\mathbf{h}_1))), && \mathbf{h}_2 \in \mathbb{R}^{B\times 256}, \\ \mathbf{z} &= \frac{\mathrm{Linear}_{256\to d}(\mathbf{h}_2)}{\|\mathrm{Linear}_{256\to d}(\mathbf{h}_2)\|_2}, && \mathbf{z} \in \mathbb{R}^{B\times d}. \end{aligned} \tag{1}$$

A contrastive loss operates on the patch representations in the batch $\mathcal{B}$. Following contrastive learning terminology, for each *anchor* patch $p_a \in \mathcal{B}$, we compute its similarities $s_{ak} = S(\mathbf{z}_a, \mathbf{z}_k)$ with each *positive* patch—a geometrically similar patch $p_k, k \in \mathcal{C}_a^+$. Here $\mathcal{C}_a^+$ is the set of "positives" for p_a. Analogously, we compute the similarity $s_{al} = S(\mathbf{z}_a, \mathbf{z}_l)$ with *negatives*—the geometrically dissimilar patches $p_l, l \in \mathcal{C}_a^-$. Together, $(p_a, \mathcal{C}_a^+, \mathcal{C}_a^-)$ constitute a *contrastive tuple*.

We use the Multiple Similarity loss, known for its robustness and flexibility:

$$\mathcal{L}_{\mathrm{MS}} = \frac{1}{m}\sum_{a=1}^{m}\left\{\frac{1}{\alpha}\log\left[1+\sum_{k\in\mathcal{C}_a^+} e^{-\alpha(s_{ak}-\lambda)}\right] + \frac{1}{\beta}\log\left[1+\sum_{l\in\mathcal{C}_a^-} e^{\beta(s_{al}-\lambda)}\right]\right\}, \tag{2}$$

where the margin $\lambda > 0$ prevents trivial solutions where the network collapses all representations to a single point. Hyperparameters α, β control how strongly the loss penalizes hard positive ($s_{ak} < \lambda$) and hard negative ($s_{al} > \lambda$) pairs.

Anomaly detection as semi-supervised contrastive learning.

Supervised contrastive learning expects that a class label y is assigned to each item $p \in \mathcal{B}$. In the context of ADL, patches with identical geometry must share the same class label. Items whose class label is the same as the label y_a of an anchor p_a form the set $\mathcal{C}_a^+$, while $\mathcal{C}_a^-$ consists of items with different labels.

Explicit labeling of surface geometry is infeasible, and existing datasets provide only per-point anomaly masks $\tilde{X}$. It is tempting—but incorrect—to use location c_i and an anomaly flag $\mathbf{1}\{P_i^b \subset \tilde{X}^b\}$ as the label $y_i = \langle c_i, \mathbf{1}\{P_i^b \subset \tilde{X}^b\}\rangle$. Even patches that originate from different centers $c_i, c_j, i \neq j$ often have the same geometry. For example, consider an object that has large areas of flat surface (e.g., a box), or an object with pronounced symmetry. Assigning different labels

to such distant but geometrically identical patches results in ill-formed tuples (see Fig. 3).

We do not explicitly define class labels for patches. Instead, we define the minimally feasible sets $\mathcal{C}_a^+$ and $\mathcal{C}_a^-$ using the inductive bias pertaining to the problem. More specifically, we note that

- all non-anomalous patches $P_i^1, \cdots, P_i^N$ centered at the location c_i have the same geometry. For the given anchor $p_a = P_i^g, g \in (1, \cdots, N)$, the remaining patches can form the set of "positives" $\mathcal{C}_a^+ = \{P_i^j : j \in (1, \cdots, N), j \neq g\}$;
- a patch $P_i^b, b \in (N+1, \cdots, M)$ extracted from one of the "bad" PCs $X^{N+1}, \cdots, X^M$ and containing an anomaly (i.e. $P_i^b \subset \tilde{X}^b$) is likely different from any other patch. It can be a negative for *any* other patch P.

We exploit such inductive bias by defining two types of contrastive tuples—fully and weakly spatially aligned, as shall be elaborated next.

First, real-world anomalies are small and only subtly deviate from a template; a representative tuple must be **fully spatially aligned (FA)**. A FA tuple contains a real anomaly and thus presents a *"strong"* learning signal. It is composed of anomalous and anomaly-free patches that belong to different objects but share the same center c_i. Let anchor p_a be a patch P_i^g extracted from an object $X^g, g \in (1, \cdots, N)$. Then $\mathcal{C}_a^+ = \{P_i^j : j \in (1, \cdots, N), j \neq g\}$. Similarly, $\mathcal{C}_a^- = \{P_i^b : b \in (N+1, \cdots, M)\}$, subject to $P_i^b \subset \tilde{X}^b$. Anomalies are sparse in the training set, so FA tuples usually have a single negative, i.e., $|\mathcal{C}_a^-| = 1$.

Figure 2 depicts a batch composed of two FA contrastive tuples, constructed from two anomaly-free PCs X^1, X^2 and two anomalous PCs X^3, X^4. The patch centers c_1 and c_2 are chosen so that each contains at least one anomalous region (shown in red). The first tuple is made up of an anchor $p_a = P_1^4$, a set of positives $\mathcal{C}_a^+ = \{P_1^1, P_1^2\}$ and a negative $\mathcal{C}_a^- = \{P_1^3\}$. Similarly, the second tuple is $(p_a = P_2^2, \mathcal{C}_a^+ = \{P_2^1, P_2^3\}, \mathcal{C}_a^- = \{P_2^4\})$.

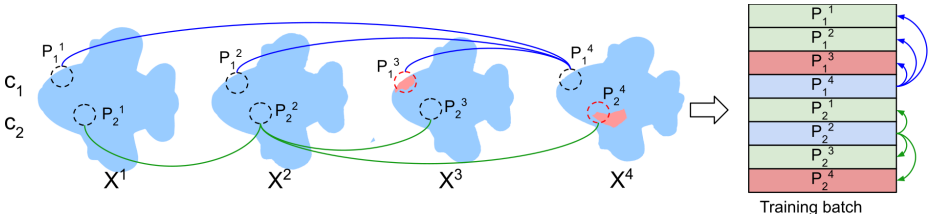

Fig. 2: Formation of a batch of fully-aligned tuples. In a batch, anchors are highlighted blue, positives – green, negatives – red. Arrows denote composition of $\mathcal{C}_a^+$, $\mathcal{C}_a^-$ sets.

However, anomalies occupy less than 2-10% of the surface of the object. This limits the amount of training data and introduces a bias—if P_i does not contain an anomaly, the geometry around c_i will not be represented in a training set. To train the encoder on patches from all locations, we define **weakly spatially aligned (WA) tuples**. In a WA tuple, only the anchor p_a and positive patches $\mathcal{C}_a^+$ originate from the same center c_i. The set of negatives $\mathcal{C}_a^-$ is the union of all negatives within batch $\mathcal{B}$ that belong to FA tuples: $\mathcal{C}_a^- = \cup_k \{\mathcal{C}_k^-\}$, where k

denotes anchors in these fully-aligned tuples. Thus, a single WA tuple contains multiple anomalies, i.e., $|\mathcal{C}_a^-| >> |\mathcal{C}_a^+| > 1$.

Figure 3 extends the previous example with a valid WA tuple (P_3^3, $\{P_3^1, P_3^2, P_3^4\}$, $\{P_1^3, P_2^4\}$) where the anchor patch P_3^3 is contrasted with all the negatives available in the batch. The tuple ($P_4^1, \{P_4^2, P_4^3, P_4^4\}, \{P_3^1\}$) is invalid here: P_4^1 and P_3^1 originate from the symmetric locations of an object and thus have identical geometry; however, P_3^1 is added to a set of "negatives" for P_4^1.

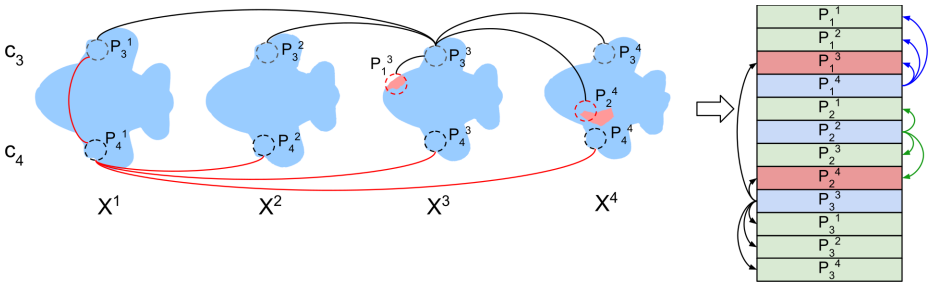

Fig. 3: A weakly-aligned contrastive tuple is added into a batch. An invalid WA tuple is shown in red; it contains identical geometries from the symmetric parts of an object.

Multi-objective learning.

To seamlessly train the encoder on FA and WA tuples, we define a *multi-objective* contrastive loss $\mathcal{L}$ as a weighted sum of losses:

$$\mathcal{L} = w_{fa}\mathcal{L}_{FA} + w_{wa}\mathcal{L}_{WA}. \tag{3}$$

$\mathcal{L}_{FA}$ and $\mathcal{L}_{WA}$ are both computed using the Multi-Similarity loss (Eq. 2). However, they represent different learning objectives, are defined over different sets $\mathcal{C}_a^+$ and $\mathcal{C}_a^-$, and have different hyperparameters α and β. $\mathcal{L}_{FA}$ is computed on the set $\mathcal{F}$ of all fully-aligned tuples in $\mathcal{B}$ and defines the "strong" primary objective: to ensure that COSSAD-3D precisely detects actual anomalies. $\mathcal{L}_{WA}$ is computed on the set of all weakly-aligned tuples $\mathcal{W}$ and defines the "weak" auxiliary objective: to improve the generalization of the encoder towards a wide variety of geometries. $\mathcal{L}_{WA}$ contrasts a template p_a with as many negatives as possible, expecting that one of them will produce a "hard negative".

Balancing "strong" and "weak" objectives is critical: if either loss dominates gradient magnitudes, the encoder may overfit to that signal while the other's contribution collapses [3]. Since the FA and WA losses differ in labeling semantics and tuple cardinality, we employ GradNorm [3] to adaptively recompute objective weights w_{fa} and w_{wa} at each step t, equalizing gradient magnitudes:

$$G_{fa} = \|\nabla_{\theta_s}(w_{fa}\mathcal{L}_{FA})\|_2, G_{wa} = \|\nabla_{\theta_s}(w_{wa}\mathcal{L}_{WA})\|_2, \bar{G} = (G_{fa} + G_{wa})/2. \tag{4}$$

$$w_{\text{fa}}^* = \arg\min_{w_{\text{fa}}} \left|G_{\text{fa}}(t) - \bar{G}(t)\right|, w_{\text{wa}}^* = \arg\min_{w_{\text{wa}}} \left|G_{\text{wa}}(t) - \bar{G}(t)\right|. \tag{5}$$

Convergence conditions on tuple composition.

The proposed $\mathcal{C}_a^-$, $\mathcal{C}_a^+$ define the minimal set of contrastive relations required for the convergence of our semi-supervised method. This follows from the gradient

expression for negative items:

$$\frac{\partial L^{(i)}}{\partial \theta} = \frac{\partial L}{\partial z_i}\frac{\partial z_i}{\partial \theta} = \left\{ \frac{w_{fa}}{m} \frac{e^{\beta\,(z_a^\top z_i - \lambda)}}{1 + e^{\beta\,(z_a^\top z_i - \lambda)}}\, z_a + \frac{w_{wa}}{m} \sum_{w \in \mathcal{W}} \frac{e^{\beta\,(z_w^\top z_i - \lambda)}}{1 + e^{\beta\,(z_w^\top z_i - \lambda)}}\, z_w \right\} \frac{\partial z_i}{\partial \theta}. \quad (6)$$

Let $p_i \in \mathcal{B}$ be a negative item. Then $p_a \in \mathcal{F}$ and $p_w \in \mathcal{W}$ are the anchors contrasted with p_i. Although $\mathcal{F}$ and $\mathcal{W}$ are disjoint in the batch $\mathcal{B}$, in the representation space they are interrelated, "joined" together as the terms of Eq. 6. The gradient of each negative z_i includes contributions from the representations z_a and z_w that correspond to FA and WA tuples, respectively, leading to a strong separation between positive and negative items in the representation space.

If a negative p_i is not contrasted with any FA anchor (e.g. p_i is a mined negative patch), the first term in Eq. 6 will be absent. Thus, the representation z_i may be pushed towards some z_a from the set of fully-aligned patches. This leads to poor separation between positives and negatives and to the loss of COSSAD-3D accuracy in our tests. Moreover, mining negatives with the MoCo-style encoder yields a gradient $\frac{\partial z_i}{\partial \theta} = 0$, leading to a collapse of representations.

3.2 Location-aware Patch Comparison

During the inference stage, we use our trained feature extractor to compare the "query" object Y^q against the set $Y^1, \cdots Y^G$ of "good" reference objects. Utilizing multiple reference PCs makes our comparison more robust to measurement noise in PCs and the stochasticity of patch sampling.

Algorithm 1 depicts the proposed computation of the location-aware patch comparison. Our algorithm starts with registering Y^q and $Y^1, \cdots Y^G$ objects into the global coordinate frame defined by Y^1 (line 1). We sample patch centers $c_1, \cdots, c_K$ using the FPS algorithm, which guarantees full and uniform coverage of PCs $Y^1, \cdots, Y^G, Y^q$ by patches (line 2). For each location c_i, we fetch patches $P_i^1, \cdots, P_i^G$ from good objects $Y^1, \cdots Y^G$, and a patch P_i^q from a query object Y^q. The number of patches K and their centers $c_i, i \in (1, \cdots, K)$ are the same across all objects.

For each "good" patch $P_i^j, j \in (1, \cdots, G)$ with a center c_i, the feature extractor computes its representations $\mathbf{z}_i^j = f_\theta(P_i^j)$. We store these representations in K local memory banks $\mathcal{M}_i$. Each local memory bank $\mathcal{M}_i$ contains representations $\mathbf{z}_i^1, \cdots, \mathbf{z}_i^G \in \mathcal{R}^d$ from the "good" patches $P_i^1, \cdots, P_i^G$ (lines 5-8).

We compare the representations $\mathbf{z}_i^q$ extracted from a query object Y^q with the corresponding sample in $\mathcal{M}_i$. However, the number of representation vectors $|\mathcal{M}_i| = G$ in a memory bank is much smaller than their dimensionality d. For example, $|\mathcal{M}_i| = 4$ for the Real3D-AD dataset, while $d = 33$ for an FPFH descriptor of a patch and $d = 64$ for COSSAD-3D. This is a problem for multivariate statistical techniques such as the Mahalanobis distance, which expects $|\mathcal{M}_i| > d$. Instead, we compare the pairwise distances between the representation vector $\mathbf{z}_i^q$ and the contents of $\mathcal{M}_i$. Let the cosine distance $\mathrm{dist}_i(q, j) = (1 - \mathbf{z_i^q} \cdot \mathbf{z_i^j}) \in \mathcal{R}$ implicitly represent the difference in the geometries of the "query" patch P_i^q and the patch P_i^j from the "reference" object Y^j. Together, $\mathcal{D}_i^q = \{\mathrm{dist}_i(q, j) : j \in (1, \cdots, G)\}$

Algorithm 1 Location-aware patch comparison

1: Register $Y^1, \cdots, Y^G, Y^q$ into the global coordinate frame of Y^1
2: Compute patch centers $(c_1, \cdots, c_K) = \text{FPS}(Y^1)$
3: **for** each patch center $c_i \in (1, \cdots, K)$ **do**
4: $\quad \mathcal{M}_i \leftarrow \emptyset, \mathcal{D}_i \leftarrow \emptyset, \mathcal{D}_i^q \leftarrow \emptyset$ ▷ Initialize local memory banks
5: $\quad$ **for** each reference object $Y^j \in (Y^1, \cdots, Y^G)$ **do**
6: $\quad\quad$ Extract patch P_i^j from Y^j with center c_i
7: $\quad\quad \mathbf{z}_i^j = f(P_i^j)$ ▷ Compute representations for a reference patch
8: $\quad\quad \mathcal{M}_i \leftarrow \mathcal{M}_i \cup \{\mathbf{z}_i^j\}$ ▷ Add representations to a local memory bank
9: $\quad$ **for** each pair $(\mathbf{z}_i^j, \mathbf{z}_i^k)$ in $\mathcal{M}_i$ **do**
10: $\quad\quad \mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{1 - \mathbf{z}_i^j \cdot \mathbf{z}_i^k\}$ ▷ Compute pairwise distances for a memory bank
11: **for** each patch P_i^q in the "query" object Y^q **do**
12: $\quad \mathbf{z}_i^q = f_\theta(P_i^q)$ ▷ Compute representations for a "query" patch
13: $\quad$ **for** each $\mathbf{z}_i^j$ in $\mathcal{M}_i$ **do**
14: $\quad\quad \mathcal{D}_i^q \leftarrow \mathcal{D}_i^q \cup \{1 - \mathbf{z}_i^j \cdot \mathbf{z}_i^q\}$
15: $\quad$ Compute patch anomaly score δ_i using (7, 8)
16: **return** Object anomaly score $\Delta = max(\delta_1, \cdots, \delta_K)$

is a set of distances between the "query" patch $\mathbf{z}_i^q$ and the sample of good patches $\mathcal{M}_i$, extracted from $Y^1, \cdots Y^G$ at the location c_i (lines 11-14).

High values of $\mathcal{D}_i^q$ can indicate the presence of an anomaly, as there is a significant difference in geometry between the "query" object and the "reference" objects in the neighborhood of the point c_i. However, $\mathcal{D}_i^q$ can be interpreted only in the context of how variable the geometries around the point c_i are in general.

Let $\text{dist}_i(j,k) = (1 - \mathbf{z}_i^j \cdot \mathbf{z}_i^k) \in \mathcal{R}$ be a pairwise distance between the features $\mathbf{z_i^k}, \mathbf{z_i^j} \in \mathcal{M}_i, k \neq j$. Then $\mathcal{D}_i = \{\text{dist}_i(j,k) : j,k \in (1, \cdots, G), k \neq j\}$ is a set of all mutual distances between the features of "reference" patches, and $|\mathcal{D}_i| = G(G-1)$. The variance $\sigma(\mathcal{D}_i)$ represents the degree of variation in the geometry of the patches extracted from our "reference" objects in the vicinity of the point c_i (lines 9-10). Large values of $\sigma(\mathcal{D}_i)$ reflect measurement noise and manufacturing tolerances. We compute the final anomaly score for a "test" patch P_i^q as Cohen's effect size δ_i [6] between the samples of pairwise distances $\mathcal{D}_i^q$ and $\mathcal{D}_i$:

$$\delta_i = \left| \frac{\overline{\mathcal{D}_i} - \overline{\mathcal{D}_i^q}}{s_i} \right|, s_i = \sqrt{\frac{(|\mathcal{D}_i| - 1)\sigma^2(\mathcal{D}_i) + (|\mathcal{D}_i^q| - 1)\sigma^2(\mathcal{D}_i^q)}{|\mathcal{D}_i| + |\mathcal{D}_i^q| - 2}}, \tag{7}$$

Finally, we compute an aggregated object-wise anomaly score as $\Delta = \max(\delta_i)$.

To localize the anomalies, we compute the anomaly score for each point $x^q \in X^q$ as the sum of the anomaly scores of the patches, weighted by the distance from x^q to the corresponding patch center $\|x^q - c_i\|$:

$$\delta(x^q) = \frac{\sum_{i=1}^K \delta_i \, \phi_i}{\sum_{i=1}^K \phi_i}, \quad \phi_i = \exp\left(-\frac{\|x^q - c_i\|^2}{2R^2} \right), \tag{8}$$

where the distance weight ϕ_i is a Gaussian kernel.

4 Experimental evaluation

Input data. We evaluate COSSAD-3D on the Anomaly-ShapeNet [10] and Real3DAD datasets [12]. Anomaly-ShapeNet contains 1720 synthetic point clouds of 52 object classes. Real3DAD contains 648 real scans of 12 object classes.

During both training and inference, we use the ICP algorithm to register PCs. ICP is prone to a suboptimal solution for highly symmetric objects. Thus, we performed multiple retries of ICP registration while rotating the input PC about its principal axes.

Encoder training. Our semi-supervised formulation must generalize across diverse 3D geometries regardless of their SO(3) orientation, which motivates a rotation-invariant backbone. We use RIConv++ [23], which defines a local coordinate system for patch p_a by performing PCA on its points' coordinates. Then RIConv++ learns a set of rotation-invariant convolutional kernels within that local coordinate system, producing a patch representation $\mathbf{h} \in \mathcal{R}^{512}$ that is insensitive to rigid transformations.

We trained the encoder on a combined dataset comprising 5385 fully aligned (FA) and 71990 weakly aligned (WA) contrastive tuples. 1670 FA and 19990 WA tuples were extracted from Real3D-AD; 3715 FA and 52000 WA tuples were extracted from Anomaly-ShapeNet. Each FA tuple contained $|\mathcal{C}_a^+| = 2$ positives and $|\mathcal{C}_a^-| = 1$ negative; each WA tuple contained $|\mathcal{C}_a^+| = 3$ positives and $|\mathcal{C}_a^-| = 40$ negatives. The training batches contained a mixture of patches from both datasets. Synthetic data spanning 52 Anomaly-ShapeNet classes improves the encoder's generalization across geometries, while real scanned data from the Real3D-AD dataset provides the inductive bias for detecting realistic anomalies [18]. Patches $p \in \mathcal{B}$ were re-sampled to a fixed length of 512 points. Positive patches were subjected to intensive augmentations, including global (per-tuple) and local (per-patch) rotations, Gaussian noise, and edge cropping.

We trained our contrastive encoder using the AdamW optimizer with lr=5e-4 and L2 regularization of 1e-5 for 120 epochs on an RTX 5090 GPU with 32 GB of VRAM. A batch was composed of 40 FA tuples and 100 WA tuples. L_{FA} and L_{WA} share the same margin $\lambda = 0.6$. For L_{FA}, we set $\alpha = 3$, $\beta = 30$ to strongly penalize misclassifications of actual anomalies. For L_{WA}, we set $\alpha = 10$ and $\beta = 10$ to encourage tight grouping of positives in a representation space. The objective weights w_{fa} and w_{wa} converged to 1.8 and 0.2, respectively, confirming $\mathcal{L}_{FA}$ as a "strong" learning objective.

Accuracy. We evaluate COSSAD-3D using 12-fold cross-validation. Each fold contained a random subset of objects from the Anomaly-ShapeNet and Real3D-AD datasets. Eleven folds were used to train the contrastive feature extractor; the 12th fold was used for evaluation. By training and validating COSSAD-3D on non-intersecting sets of object shapes, we verify how COSSAD-3D generalizes to unseen object geometries.

We use the Area Under the Receiver Operating Characteristic (AUROC) metric to measure the accuracy of COSSAD-3D. For each object, we calculate both the object-wise AUROC (o-ROC) and the point-wise AUROC (p-ROC); we report the mean accuracy over 5 inference runs per object.

The proposed COSSAD-3D is significantly more accurate than the existing methods. On the most representative dataset, Real3D-AD, COSSAD-3D achieves an average o-ROC of 91.2% compared to 82.9% by the best known algorithm, PointCore. Furthermore, our approach localizes defects with high accuracy, achieving a mean p-ROC of 95.0% compared to 92.3% reported by Simple3D. For the Anomaly-ShapeNet dataset, COSSAD-3D shows an average o-ROC =94.9% and p-ROC =96.1%, which is significantly higher than other known methods. Figure 4 summarizes the COSSAD-3D accuracy in all Anomaly-ShapeNet objects.

Explicit accuracy comparison may be complicated by the fact that the strongest prior baselines [12], [5], [4], [24] are unsupervised, and only PO3AD [22] uses synthetic anomalies to train separate supervised models for each object. Instead, COSSAD-3D is a semi-supervised method that leverages the anomaly annotations already provided in the evaluation splits of datasets.

COSSAD-3D performs notably worse on four "difficult" objects in Real3D-AD: "Seahorse", "Shell", "Starfish", and "Toffees". Qualitatively, the shapes and surface textures of these objects differ significantly from those in the rest of the training set, potentially leading to OOD patch geometries.

Generalization to unseen defect types cannot be reliably assessed, as the Real3D-AD and Anomaly-ShapeNet datasets feature limited defect diversity, with bulges and sinks accounting for 98% and 77% of all defects, respectively. This limitation affects the entire field of 3D anomaly detection, since even unsupervised defect detection methods can be implicitly biased towards these defect types.

Table 1: Accuracy on Real3D-AD dataset, reported as o-ROC/p-ROC. The best result is in **bold**, second best - underlined.

Method → Object	Reg3D-AD [12]	PO3AD [22]	Simple3D [5]	MC3D-AD [4]	PointCore [24]	COSSAD-3D
Airplane	71.6/63.1	80.4/-	76.5/88.1	85.0/62.8	66.0/60.8	**91.8**/**96.8**
Candybar	82.7/72.2	78.5/-	85.1/96.2	77.8/73.6	97.6/76.0	**100**/**98.1**
Car	69.7/71.8	65.4/-	**98.1**/**99.2**	74.9/81.9	86.6/70.6	96.5/98.0
Chicken	**85.2**/67.6	68.6/-	82.6/86.1	71.5/64.0	84.1/78.0	85.0/**88.9**
Diamond	90.0/83.5	80.1/-	**100**/**99.0**	95.5/94.2	96.3/81.0	**100**/97.7
Duck	58.4/50.3	82.0/-	78.7/96.6	83.1/82.2	68.4/71.2	**93.9**/**97.1**
Fish	91.9/85.2	85.9/-	91.2/**99.6**	91.2/90.6	99.2/78.2	**99.3**/96.3
Gemstone	41.7/54.5	69.3/-	70.4/**97.3**	56.0/45.8	53.4/51.5	**98.9**/97.1
Seahorse	76.2/81.7	75.6/-	93.0/94.2	90.1/95.0	**97.3**/84.1	85.6/**96.3**
Shell	35.8/81.1	80.0/-	85.1/**97.6**	51.5/47.1	**86.1**/78.1	73.4/88.1
Starfish	50.6/71.7	75.8/-	69.5/85.8	76.6/69.0	65.2/73.6	**83.4**/**94.1**
Toffees	68.5/75.9	77.1/-	88.9/**96.8**	78.3/93.4	**92.9**/74.5	86.5/91.8
Mean	70.4/70.5	76.7/83.6	80.4/92.3	78.2/76.8	82.9/73.1	**91.2**/**95.0**

Ablation studies. To measure the contribution of our key innovations—contrastive encoder, multi-objective learning, and spatially-aware patch comparison—we selectively replaced these components with SOTA analogs and measured the o-ROC and p-ROC on the Real3D-AD dataset (see Table 2).

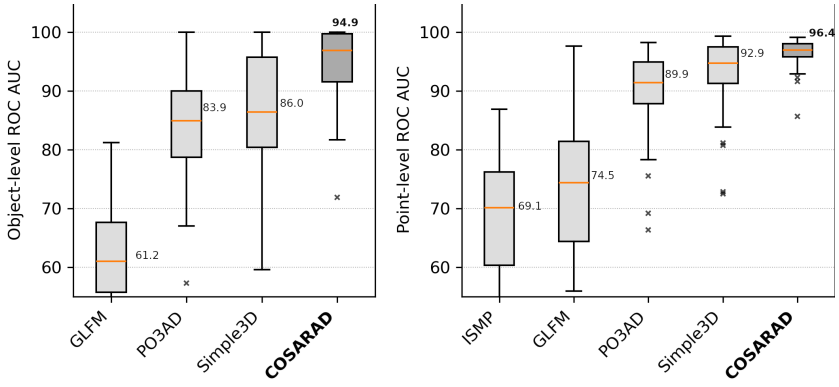

Fig. 4: Accuracy on the Anomaly-ShapeNet dataset

Replacing learned representations with FPFH descriptors reduced the mean o-ROC from 91.2% to 75.9%, approaching Reg3D-AD baseline. These results demonstrate that the contrastive feature extractor is a key factor in achieving high anomaly detection accuracy.

To evaluate the contribution of multi-objective learning (MOL), we trained COSSAD-3D using only fully aligned patches ("FA only"), reducing it to a fully supervised contrastive method. The overall o-ROC dropped to 84.1%, supporting our hypothesis that semi-supervised contrastive learning makes the encoder more robust to OOD geometries.

Training on weakly aligned patches alone ("WA only") reduced the mean o-ROC to 87.6% — higher than the model trained using a "strong" FA-only objective. Although surprising, this observation aligns with prior work on weakly-supervised contrastive methods [25], [19]: the "weak" objective, while noisier, operates on a larger and more diverse set of geometries.

Replacing the spatially aware comparison with PatchCore's reference implementation reduced the mean o-ROC to 81.8%, confirming the strong contribution of spatial awareness to detection accuracy.

Finally, we assessed COSSAD-3D's sensitivity to registration errors by perturbing the query object Y^q with random rotations $R' = \pm 3°, \pm 5°$ and translations $t' = \pm 3\%, \pm 5\%$ of the object's dimensions. At $(R' = \pm 3°, t' = \pm 3\%)$, o-ROC dropped slightly to 88.3%, likely because similar perturbations were used as augmentations during encoder training. At $(R' = \pm 5°, t' = \pm 5\%)$, o-ROC dropped to 83.4%, underscoring registration's role for spatially-aware anomaly detection.

5 Conclusion

We presented COSSAD-3D, a 3D point cloud anomaly detection algorithm designed for high-accuracy precision manufacturing applications such as aerospace. By combining a contrastive feature extractor with spatially aware patch comparison, COSSAD-3D establishes a new accuracy benchmark on the Real3D-AD and Anomaly-ShapeNet datasets. To improve accuracy on out-of-distribution

Table 2: Ablation studies. ↓ indicates a significant ($p < 0.05$) decrease in per-object accuracy compared to baseline. ↑ indicates a significant improvement.

Object	Baseline	FPFH	FA only	WA only	PatchCore	Reg. ±3	Reg. ±5
Airplane	91.8/96.8	$73.1^{\downarrow}$/$92.3^{\downarrow}$	91.1 /96.7	$88.5^{\downarrow}$/$96.1^{\downarrow}$	$81.0^{\downarrow}$/$98.0^{\uparrow}$	$79.9^{\downarrow}$/$94.8^{\downarrow}$	$75.9^{\downarrow}$/$92.9^{\downarrow}$
Candybar	100/98.1	$89.4^{\downarrow}$/98.1	100 /$98.4^{\uparrow}$	100 /97.5	99.8 /$99.5^{\uparrow}$	$99.2^{\downarrow}$/$98.0^{\downarrow}$	$96.3^{\downarrow}$/$97.6^{\downarrow}$
Car	96.5/98.0	$81.8^{\downarrow}$/$94.0^{\downarrow}$	$93.3^{\downarrow}$/97.9	96.4 /$97.9^{\downarrow}$	$64.7^{\downarrow}$/$97.6^{\downarrow}$	$88.3^{\downarrow}$/$95.9^{\downarrow}$	$83.6^{\downarrow}$/$91.8^{\downarrow}$
Chicken	85.0/88.9	87.5 /88.2	88.9 /86.2	83.7 /88.6	86.7 /$94.9^{\uparrow}$	85.8 /$86.6^{\downarrow}$	$81.7^{\downarrow}$/86.2
Diamond	100/97.7	$95.5^{\downarrow}$/$97.1^{\downarrow}$	99.8 /$97.0^{\downarrow}$	100 /97.1	99.5 /$98.7^{\uparrow}$	100 /97.2	99.9 /$97.2^{\downarrow}$
Duck	93.9/97.1	$80.0^{\downarrow}$/$92.2^{\downarrow}$	$86.8^{\downarrow}$/$96.4^{\downarrow}$	92.5 /$96.8^{\downarrow}$	91.3 /$99.5^{\uparrow}$	$89.1^{\downarrow}$/96.9	$80.9^{\downarrow}$/$95.3^{\downarrow}$
Fish	99.3/96.3	$88.3^{\downarrow}$/$95.7^{\downarrow}$	$91.7^{\downarrow}$/$94.4^{\downarrow}$	98.8 /96.3	$90.1^{\downarrow}$/$98.5^{\uparrow}$	$97.1^{\downarrow}$/96.2	$91.5^{\downarrow}$/$95.4^{\downarrow}$
Gemstone	98.9/97.1	$70.1^{\downarrow}$/$90.9^{\downarrow}$	99.2 /$97.5^{\uparrow}$	97.5 /96.9	91.6$^{\downarrow}$/98.2$^{\uparrow}$	98.4 /97.3	97.8 /97.1
Seahorse	85.6/96.3	$60.7^{\downarrow}$/$87.4^{\downarrow}$	$58.5^{\downarrow}$/$89.4^{\downarrow}$	$70.2^{\downarrow}$/$95.6^{\downarrow}$	$75.8^{\downarrow}$/$93.5^{\downarrow}$	$80.3^{\downarrow}$/$94.9^{\downarrow}$	$76.0^{\downarrow}$/$92.2^{\downarrow}$
Shell	73.4/88.1	$49.4^{\downarrow}$/$72.2^{\downarrow}$	$55.2^{\downarrow}$/87.9	$63.7^{\downarrow}$/84.0	$52.0^{\downarrow}$/86.5	70.7 /88.9	67.2 /86.7
Starfish	83.4/94.1	$58.2^{\downarrow}$/$77.2^{\downarrow}$	$60.5^{\downarrow}$/$91.5^{\downarrow}$	$76.2^{\downarrow}$/$89.3^{\downarrow}$	84.8 /$95.7^{\uparrow}$	82.9 /94.0	79.9 /92.2
Toffees	86.5/91.8	77.3 /$89.2^{\downarrow}$	84.2 /90.4	84.6 /89.8	$64.1^{\downarrow}$/$96.6^{\uparrow}$	87.6 /89.9	80.1 /$86.7^{\downarrow}$
Mean	91.2/95.0	75.9 /89.5	84.1 /93.6	87.6 /93.8	81.8 /96.4	88.3 /94.2	83.4 /92.6

geometries, we formulated anomaly detection as a semi-supervised contrastive learning problem and established minimal convergence conditions on tuple composition. This formulation expands the effective training set and improves the encoder's generalization to unseen objects. We further strengthened spatially aware detection by comparing each test object against multiple reference shapes.

Additional gains may come from jointly reasoning over neighboring patches and the spatial arrangement of their centers. We are developing a graph-based comparison module to exploit both modalities.

Our source code is available at `https://github.com/alextarvo/cossad`.

References

1. Bergmann, P., Sattlegger, D.: Anomaly detection in 3d point clouds using deep geometric descriptors. In: Winter Conf. on Applications of Computer Vision (2023)
2. Chen, T., Kornblith, S., Norouzi, M., et al.: A simple framework for contrastive learning of visual representations. In: Intl. Conf. on Machine Learning (2020)
3. Chen, Z., Badrinarayanan, V., Lee, C.Y., et al.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: Intl. Conf. on Machine Learning (2018)
4. Cheng, J., Gao, C., Zhou, J., et al.: Mc3d-ad: A unified geometry-aware reconstruction model for multi-category 3d anomaly detection. In: Intl. Joint Conf. on Artificial Intelligence (2025)
5. Cheng, Y., Sun, Y., Zhang, H., et al.: Towards high-resolution 3d anomaly detection: A scalable dataset and real-time framework for subtle industrial defects (2025), `https://arxiv.org/abs/2507.07435`
6. Cohen, J.: Statistical Power Analysis for the Behavioral Sciences. Lawrence Erlbaum Associates, Hillsdale, NJ, 2nd edn. (1988)
7. He, K., Fan, H., Wu, Y., et al.: Momentum contrast for unsupervised visual representation learning. In: Conf. on Computer Vision and Pattern Recognition (2020)
8. Horwitz, E., Hoshen, Y.: Back to the feature: Classical 3d features are (almost) all you need for 3d anomaly detection (2022), `https://arxiv.org/abs/2203.05550`

9. Khosla, P., Teterwak, P., Wang, C., et al.: Supervised contrastive learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
10. Li, W., Xu, X., Gu, Y., et al.: Towards scalable 3d anomaly detection and localization: A benchmark via 3d anomaly synthesis and a self-supervised learning network (2023), https://arxiv.org/abs/2311.14897
11. Liang, H., Xie, G., Hou, C., et al.: Look inside for more: internal spatial modality perception for 3d anomaly detection. In: Conf. on Artificial Intelligence (2025)
12. Liu, J., Xie, G., Chen, R., et al.: Real3d-ad: A dataset of point cloud anomaly detection (2023), https://arxiv.org/abs/2309.13226
13. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. In: Advances in Neural Information Processing Systems (2018)
14. Roth, K., Pemula, L., Zepeda, J., et al.: Towards total recall in industrial anomaly detection. In: Conf. on Computer Vision and Pattern Recognition (2022)
15. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: IEEE Intl. Conf. on Robotics and Automation (2009)
16. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Conf. on Computer Vision and Pattern Recognition (2015)
17. Sun, Y., Cheng, C., Zhang, Y., et al.: Circle loss: A unified perspective of pair similarity optimization. In: Conf. on Computer Vision and Pattern Recognition (2020)
18. Tremblay, J., Prakash, A., Acuna, D., et al.: Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In: Conf. on Computer Vision and Pattern Recognition Workshops (2018)
19. Tsai, Y.H.H., Li, T., Liu, W., et al.: Learning weakly-supervised contrastive representations (2022), https://arxiv.org/abs/2202.06670
20. Wang, X., Han, X., Huang, W., et al.: Multi-similarity loss with general pair weighting for deep metric learning. In: Conf. on Computer Vision and Pattern Recognition (2019)
21. Yang, F., Wu, K., Zhang, S., et al.: Class-aware contrastive semi-supervised learning. In: Conf. on Computer Vision and Pattern Recognition (2022)
22. Ye, J., Zhao, W., Yang, X., et al.: Po3ad: Predicting point offsets toward better 3d point cloud anomaly detection. In: Conf. on Computer Vision and Pattern Recognition (June 2025)
23. Zhang, C., Yang, Y., Wang, Y., et al.: Riconv++: Rotation-invariant convolution for 3d point clouds. Tran. on Pattern Analysis and Machine Intelligence (2023)
24. Zhao, B., Xiong, Q., Zhang, X., et al.: Pointcore: Efficient unsupervised point cloud anomaly detector using local-global features (2024), https://arxiv.org/abs/2403.01804
25. Zheng, M., Wang, F., You, S., et al.: Weakly supervised contrastive learning. In: International Conference on Computer Vision (2021)
26. Zhou, Z., Wang, L., Fang, N., et al.: R3d-ad: Reconstruction via diffusion for 3d anomaly detection. In: ECCV 2024: 18th European Conf. (2024)
27. Zhu, H., Xie, G., Hou, C., et al.: Towards high-resolution 3d anomaly detection via group-level feature contrastive learning. In: ACM Intl. Conf. on Multimedia (2024)
28. Zou, Y., et al.: M3dm: Multi-modal 3d anomaly detection via hybrid fusion. In: Conf. on Computer Vision and Pattern Recognition (2023)