FACETS: Cross-Granularity Vision–Language Modeling for 3D Anomaly Detection

Yuchuan Li
Queen's University
Kingston, Canada
li.yuchuan@queensu.ca

Jae-Mo Kang
Kyungpook National University
Daegu, South Korea
jmkang@knu.ac.kr

Il-Min Kim*
Queen's University
Kingston, Canada
ilmin.kim@queensu.ca

Abstract

Three-dimensional (3D) anomaly detection underpins quality assurance in advanced manufacturing and engineering, capturing subtle defects and deviations that 2D inspection struggles to resolve due to occlusion, viewpoint, and appearance confounds. Most existing 3D anomaly detection methods either rely on reconstruction errors or stored normal representations to detect anomalies. While both paradigms have driven substantial progress, they share a fundamental limitation: both operate entirely in the visual domain, missing the rich semantics encoded in natural language, which also typically constrains them to per-category models. We propose FACETS, a framework that leverages a 3D vision–language model for 3D anomaly detection, moving beyond reconstruction- and memory-bank-based methods. The key idea is to explicitly retain native point-level features that enable reasoning at both patch and point granularities, and further leverage category-level 3D–text grounding and cross-granularity geometric modeling along two complementary axes, linguistic semantics and geometric saliency, so that coarse semantic cues and fine-grained geometric details jointly support anomaly detection. FACETS enables unified multi-category anomaly detection, avoiding the per-category models used by most prior methods. Notably, we provide a mathematical analysis of the within-patch component of our point-level anomaly score, offering design guidance for our loss function in anomaly localization. Extensive experiments on popular benchmarks reveal that FACETS sets a new state of the art in average AUROC across datasets and metrics for 3D anomaly detection, substantially and consistently outperforming existing methods. Code is available at https://github.com/infinitr0us/FACETS.

1 Introduction

Three-dimensional (3D) anomaly detection is essential for ensuring quality and reliability in advanced manufacturing, precision machining, and automated inspection. In these settings, subtle surface defects and structural deviations cannot be reliably identified with 2D imagery, as occlusion, viewpoint, lighting, and appearance confounds routinely mask geometry-intrinsic faults [10]. While early benchmarks [2, 4] represent 3D geometry through RGB-D scans, recent work has shifted toward high-resolution point clouds as the primary data modality [23, 20], as point clouds better preserve high-precision 3D geometric details of objects. The 3D anomaly detection task demands reasoning at two complementary granularities: object-level detection, which determines whether a point cloud as a whole is anomalous, and point-level localization, which identifies precisely which regions deviate from normal geometry. Both are necessary to ensure robust anomaly detection performance.

*Corresponding author.

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

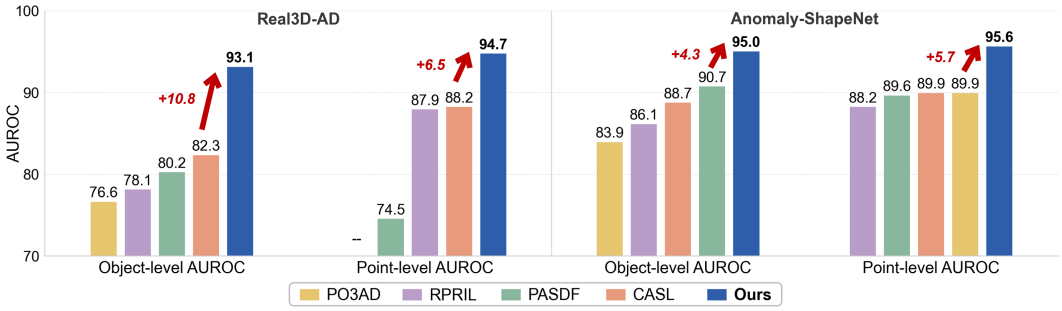


Figure 1: AUROC (averaged over all classes in each dataset, ↑) comparison on Real3D-AD [23] and Anomaly-ShapeNet [20] benchmarks for 3D anomaly detection. FACETS achieves the best average AUROC across datasets and metrics. Bars of each group are sorted in ascending order.

Existing methods for 3D anomaly detection fall into two dominant paradigms: (i) reconstruction-based methods [20, 50, 43, 47, 46] train models to reconstruct normal samples and detect anomalies via reconstruction errors, and (ii) memory-bank-based methods [15, 40, 33, 6, 23, 51, 22, 45] store representative features of normal samples and detect anomalies by measuring distances to the stored distribution. While these two paradigms have driven substantial progress [47, 45, 46, 43], they share a fundamental limitation: both operate entirely in the visual domain, treating class labels as one-hot encodings and missing rich semantics encoded in natural language. This purely visual approach also typically constrains these methods to per-category models, as they lack a shared representation space.

Recent advances in 3D Vision–Language Models (VLMs) offer a compelling opportunity to address this limitation. In particular, ULIP-2 [42], built on the Point-BERT [44] 3D encoder, has demonstrated strong 3D understanding by aligning point cloud, image, and text modalities in a shared embedding space through pre-training. This tri-modal alignment provides both a semantic scaffold that can relate geometric features to textual descriptions and a rich pre-trained representation of 3D objects. However, directly applying ULIP-2 to 3D anomaly detection is non-trivial: the tokenizer of its 3D encoder discards fine-grained point-level 3D geometric details through max-pooling within each local patch, and the standard output aggregation is geometry-agnostic, treating spatially and semantically distinct patches equally. To date, only a few works have explored the use of a VLM for 3D anomaly detection. PointAD [49] relies on a 2D VLM (i.e., CLIP [31]) through multi-view rendering: it projects point clouds into multiple 2D images and aggregates pixel-level predictions, which is not 3D-spatially aware, requires multiple forward passes through the image encoder, and struggles to detect 3D anomalies that are not visible in 2D projections. PLANE [38] instead adapts ULIP-2 with learnable dual prompts but does not retain the native point-level features discarded by its tokenizer, and BTP [18] addresses the zero-shot setting.

In this work, we propose FACETS (**F**ine-grained **A**nomaly detection via **C**ross-granularity **E**nriched **T**extual **S**emantics). FACETS is distinguished by building on the native point-level features that the tokenizer discards: it retains them as the foundation of the entire framework and grounds them, together with patch features, to category-level text, moving beyond existing reconstruction- and memory-bank-based methods. FACETS fully exploits the representational capacity of ULIP-2 by introducing a series of cross-granularity innovations that capture fine-grained 3D geometric details, integrate category-level 3D–text grounding, and retain subtle point-level 3D geometric details of the 3D object.

Specifically, FACETS enriches the 3D representation along two complementary axes, linguistic semantics and geometric saliency, at both patch and point granularities. At the patch granularity, it grounds 3D features in category-level textual semantics. At the point granularity, it first recovers subtle geometric cues otherwise discarded during tokenization, then grounds them analogously in textual semantics. Finally, holistic–local reasoning fuses representations across both granularities to capture fine-grained 3D geometric details. Together, these modules facilitate unified multi-category anomaly detection, which is challenging for most prior works.

Figure 1 shows that FACETS sets a new state of the art in average AUROC across datasets and metrics, consistently outperforming all reconstruction- and memory-bank-based methods.

Our key contributions are summarized as follows:

- We propose FACETS, a framework that leverages a 3D vision–language model for 3D anomaly detection, moving beyond existing reconstruction- and memory-bank-based approaches.
- Subtle point-level 3D geometric cues are recovered by aggregating native pre-max-pooling point-level features via point geometric retention between point representations and patch embeddings, which provides the substrate for the entire framework.
- Patch and consolidated-point representations are grounded to fixed category text anchors in ULIP-2's joint space by patch and consolidated-point linguistic grounding; controlled ablations show a meaningful but incremental gain over learnable category conditioning.
- Fine-grained 3D geometric details are captured by combining the consolidated point representations, `[CLS]` embedding, and patch embeddings through holistic–local attention.
- Mathematical analysis of the within-patch component of our point-level anomaly score provides design guidance for point-level anomaly scoring. The mathematical result is well-aligned with empirical experiments.
- FACETS sets a new state of the art in average AUROC on popular benchmarks for object- and point-level 3D anomaly detection, consistently outperforming existing methods.

2 Related Work

We briefly review relevant literature in this section and include a detailed discussion in Appendix A.

2.1 3D Anomaly Detection

Reconstruction-based methods detect anomalies via errors in reconstructing normal samples, based on the premise that a model trained exclusively on normal samples will fail to faithfully reconstruct anomalous regions. Representative directions include self-supervised reconstruction with synthetic anomalies (IMRNet [20]), diffusion-based iterative denoising from corrupted inputs (R3D-AD [50]), pseudo-abnormal offset learning with surface-normal-guided augmentation (PO3AD [43]), pose-aware signed distance functions that disentangle pose and shape via canonical alignment (PASDF [47]), and curvature-guided reconstruction with multi-scale curvature (CASL [46]).

Memory-bank-based methods store representative features extracted from normal training samples and score anomalies by the distance to the stored feature distribution, leveraging feature extractors instead of a reconstruction network. This line includes hand-crafted 3D descriptors (BTF [15]), coreset-based pre-trained features (PatchCore [33]), multimodal 2D–3D fusion (M3DM [40]), registration-based alignment for pose-robust feature matching (Reg3D-AD [23], RPRIL [45]), pseudo-multimodal features from point clouds alone (CPMF [6]), group-level feature contrastive learning for high-resolution point clouds (Group3AD [51]), and internal-viewpoint spatial features (ISMP [22]).

Despite significant progress, both paradigms operate entirely in the visual domain without leveraging semantics encoded in natural language, typically restricting them to per-category models.

2.2 Vision–Language Models for 3D Understanding

2D Vision–Language Models (VLMs) such as CLIP [31] have dominated various 2D-vision tasks. Building on this success, ULIP [41] and ULIP-2 [42] align a 3D encoder, e.g., Point-BERT [44], with CLIP's image–text space. Despite this progress, only a few works apply VLMs to 3D anomaly detection. PointAD [49] is 2D-VLM-based: it relies on a 2D VLM backbone via multi-view rendering, making it not natively 3D-aware, requiring multiple image-encoder passes, and potentially missing anomalies invisible in rendered views. Native 3D-VLM-based methods operate directly on point clouds: PLANE [38] adapts ULIP-2 with learnable dual prompts trained on synthetic pseudo-anomalies, and BTP [18] studies zero-shot 3D anomaly detection with point–language models. FACETS also directly leverages a 3D VLM for native 3D reasoning, but, unlike PLANE, it retains the native pre-max-pooling point features of the tokenizer and grounds them to category-level text, while exploiting ULIP-2's shared semantic space for unified multi-category training.

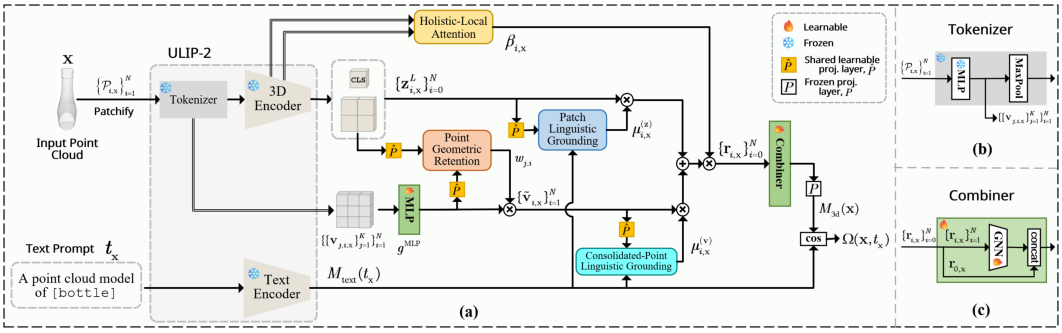


Figure 2: **Overview of our FACETS framework.** The model exploits the representational capacity of ULIP-2 by extracting and enriching cross-granularity 3D geometric features through holistic–local attention, patch linguistic grounding, point geometric retention, and consolidated-point linguistic grounding via 3D vision–language reasoning. A more detailed diagram is available in Appendix B.

3 Proposed Method

We present FACETS, a framework that leverages a 3D VLM for 3D anomaly detection. We start from two limitations of a vanilla ULIP-2 [42]: its tokenizer does not retain native pre-max-pooling point features downstream, and its standard output aggregation does not explicitly preserve region saliency or relative patch geometry. To address them, FACETS fully exploits the representational capacity of the 3D VLM by retaining subtle point-level geometric cues that are lost during tokenization, enriching the 3D–textual semantics through category-level 3D–text grounding, and integrating fine-grained geometric details for the 3D object.

3.1 Architecture of the Proposed Method

Our proposed FACETS framework, shown in Figure 2, builds on a frozen ULIP-2 [42] backbone and introduces four novel and mutually complementary mechanisms: (i) **holistic–local attention** to capture the 3D geometric details, (ii) **patch linguistic grounding** to enrich textual semantics, (iii) **point geometric retention** to recover point-level 3D geometric cues, and (iv) **consolidated-point linguistic grounding** to integrate textual semantics. Appendix B.1 summarizes the design rationale.

3.2 Preliminaries: 3D Encoding and Text Encoding

In the training phase of 3D anomaly detection, following the standard protocol in existing works, we assume access only to normal samples. Each input sample $\mathbf{x} \in \mathbb{R}^{N_p \times 3}$, which is a point cloud of N_p points, is processed by the 3D encoder to produce the 3D embedding. First, each input point cloud $\mathbf{x}$ is divided into N point patches $\{\mathcal{P}_{i,\mathbf{x}}\}_{i=1}^{N}$, each with K points, by selecting N patch centers with farthest point sampling and grouping the K nearest neighbors of each center (patches may overlap; Appendix F). Then, a tokenizer (Mini-PointNet [29]) projects each patch into an initial patch embedding. These initial embeddings are then processed by the 3D encoder, which is a Point-BERT composed of L layers. The output of layer $l = 1, \cdots, L$ is denoted as $Z^l_{\mathbf{x}} = [\mathbf{z}^l_{0,\mathbf{x}}, \mathbf{z}^l_{1,\mathbf{x}}, \cdots, \mathbf{z}^l_{N,\mathbf{x}}] \in \mathbb{R}^{d\times(N+1)}$, where $\mathbf{z}^l_{0,\mathbf{x}} \in \mathbb{R}^d$ is the `[CLS]` embedding and $\{\mathbf{z}^l_{i,\mathbf{x}}\}_{i=1}^{N} \in \mathbb{R}^{d\times N}$ are the N patch embeddings at layer l, each of dimension d. In the standard setup of ULIP-2, the final 3D representation is obtained by concatenating the `[CLS]` embedding $\mathbf{z}^L_{0,\mathbf{x}}$ with channel-wise max-pooling over N patch embeddings $\{\mathbf{z}^L_{i,\mathbf{x}}\}_{i=1}^{N}$ at the last layer L of Point-BERT as $[\mathbf{z}^L_{0,\mathbf{x}}; \max_{i=1}^{N}(\mathbf{z}^L_{i,\mathbf{x}})] \in \mathbb{R}^{2d}$, which is then projected to the joint 3D–text embedding space of dimension $\bar{d}$ using a projection matrix $P \in \mathbb{R}^{\bar{d}\times 2d}$. For the text part, the prompt for $\mathbf{x}$ is denoted as $t_{\mathbf{x}}$, which is processed by the text encoder to construct the text embedding $M_{\text{text}}(t_{\mathbf{x}})$.

However, this max-pooling aggregation in the 3D encoder treats spatially and semantically distinct patches equally, which often fails to fully exploit the fine-grained 3D geometric features preserved in the patch embeddings $\{\mathbf{z}^L_{i,\mathbf{x}}\}_{i=1}^{N}$. Such fine-grained 3D geometric features are essential for 3D anomaly detection. To address these limitations, we introduce four new, effective, and mutually

complementary mechanisms that enrich the 3D representation along two axes: linguistic semantics and 3D geometric saliency, and at two granularities: patch-level and point-level.

3.3 Holistic–Local Attention for 3D Geometric Saliency

To capture the richer 3D geometric information in patch embeddings $\{\mathbf{z}_{i,\mathbf{x}}^{L}\}_{i=1}^{N}$, our method jointly leverages the [CLS] output embedding and all patch output embeddings through an effective combination mechanism. Specifically, we construct two sets of weights, $\{\beta_{i,\mathbf{x}}\}_{i=0}^{N}$ and $\{\mu_{i,\mathbf{x}}^{(\mathbf{z})}\}_{i=0}^{N}$, which encode two complementary cues: (i) the 3D geometric relevance between the holistic [CLS] representation and local geometric details of input sample $\mathbf{x}$, and (ii) the linguistic grounding of the 3D geometric appearance using textual description of input sample $\mathbf{x}$. The first cue is described below, and the second is detailed in Sec. 3.4.

To measure the 3D geometric relevance between the holistic representation and local 3D geometric details of $\mathbf{x}$, we introduce a holistic–local attention mechanism at the 3D encoder. Specifically, we construct attention coefficients $\alpha_{i,0}^{l,h}$ at head h of layer l, which is determined as follows:

$$\alpha_{i,0}^{l,h} = \text{softmax}_i\left(\frac{1}{\sqrt{d_h}}\left(\mathbf{k}_i^{l,h}\right)^{\top}\mathbf{q}_0^{l,h}\right), \tag{1}$$

where $\mathbf{q}_0^{l,h} = (W_Q^{l,h})^{\top}(\mathbf{z}_{0,\mathbf{x}}^{l-1} + \mathbf{p}_0)$ is the query of the [CLS] token, $W_Q^{l,h}$ is the projection matrix to produce query at head h of layer l, and $\mathbf{p}_0$ is the positional embedding for the [CLS] token. $\mathbf{k}_i^{l,h} = (W_K^{l,h})^{\top}(\mathbf{z}_{i,\mathbf{x}}^{l-1} + \mathbf{p}_i)$ is the key of patch token i, $W_K^{l,h}$ is the projection matrix to produce key at head h of layer l, and $\mathbf{p}_i$ is the positional embedding for patch token i. $d_h = d/H$ is the per-head dimension and H is the number of heads per layer of Point-BERT in 3D encoder. Notably, $\alpha_{i,0}^{l,h}$ quantifies how much the [CLS] token $\mathbf{z}_{0,\mathbf{x}}^{l-1}$ attends to each patch token $\mathbf{z}_{i,\mathbf{x}}^{l-1}$ in the 3D encoder.

To capture comprehensive 3D geometric cues across the entire encoding process in Point-BERT of the 3D encoder, we aggregate $\alpha_{i,0}^{l,h}$ across all heads and all layers to obtain the final 3D relevance weight $\beta_{i,\mathbf{x}}$:

$$\beta_{i,\mathbf{x}} = \frac{1}{LH}\sum_{l=1}^{L}\sum_{h=1}^{H}\alpha_{i,0}^{l,h}. \tag{2}$$

As $\beta_{i,\mathbf{x}}$ is aggregated across all layers and heads of Point-BERT in the 3D encoder, it captures region-level geometric importance determined by each patch's spatial position and content, which is a property of the 3D region itself rather than of any particular feature representation of that region. Consequently, $\beta_{i,\mathbf{x}}$ serves as a shared saliency weight for both the patch output embeddings and the consolidated point representations (Sec. 3.5). Detailed analysis is provided in Appendix C.

3.4 Patch Linguistic Grounding for 3D Geometric Semantics

To further enhance the combining by $\beta_{i,\mathbf{x}}$, we introduce an additional set of weights $\{\mu_{i,\mathbf{x}}^{(\mathbf{z})}\}_{i=0}^{N}$ to integrate linguistic grounding for output embeddings $\{\mathbf{z}_{i,\mathbf{x}}^{L}\}_{i=0}^{N}$ through textual description of $\mathbf{x}$. Specifically, we measure the cosine similarity between the output embeddings of the 3D encoder and the text embedding of the text encoder as follows:

$$\mu_{i,\mathbf{x}}^{(\mathbf{z})} = \max\left(\cos\left(\hat{P}\mathbf{z}_{i,\mathbf{x}}^{L},\, M_{\text{text}}(t_{\mathbf{x}})\right),\, 0\right), \tag{3}$$

where $\hat{P} \in \mathbb{R}^{\bar{d}\times d}$ is a learnable projection layer. FACETS uses deterministic category templates for category-level grounding, rather than reasoning over textual descriptions of anomaly patterns.

Together, $\beta_{i,\mathbf{x}}$ and $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$ enrich the combination of output embeddings $\{\mathbf{z}_{i,\mathbf{x}}^{L}\}_{i=0}^{N}$. However, the tokenizer of the 3D encoder uses max-pooling to compress K point features into a single patch embedding (Fig. 2(b)), which may discard fine-grained point-level 3D geometric details crucial for anomaly localization. We next describe how we retain these lost details and enrich them analogously.

3.5 Point Geometric Retention for Fine-grained 3D Geometric Details

To retain the geometric information lost in max-pooling of the 3D encoder's tokenizer, we extract the pre-max-pooling representation of point j in patch i of $\mathbf{x}$ from the tokenizer, denoted as $\mathbf{v}_{j,i,\mathbf{x}}$. Then, we introduce a lightweight MLP $g^{\text{MLP}}(\cdot)$ to project them into the same space as the output embeddings of the 3D encoder. To capture the rich point-level 3D geometric details of all points within a patch, our method jointly leverages all point representations in the patch via an effective combination mechanism. Specifically, we introduce a set of weights $\{w_{j,i,\mathbf{x}}\}_{j=1}^{K}$ that encodes the alignment between the 3D appearance of point j and patch i by the cosine similarity between the projected point representations and the patch output embeddings of the 3D encoder as follows:

$$w_{j,i,\mathbf{x}} = \max\left(\cos\left(\hat{P}g^{\text{MLP}}(\mathbf{v}_{j,i,\mathbf{x}}),\ \hat{P}\mathbf{z}_{i,\mathbf{x}}^{L}\right), 0\right). \tag{4}$$

To obtain an integrated representation for patch i from its point-level representations, we combine all point representations $\{\mathbf{v}_{j,i,\mathbf{x}}\}_{j=1}^{K}$ in patch i using weights $\{w_{j,i,\mathbf{x}}\}_{j=1}^{K}$, which is computed as:

$$\tilde{\mathbf{v}}_{i,\mathbf{x}} = \sum_{j=1}^{K} w_{j,i,\mathbf{x}} \cdot g^{\text{MLP}}(\mathbf{v}_{j,i,\mathbf{x}}) \in \mathbb{R}^{d}. \tag{5}$$

To further capture the rich 3D geometric details in consolidated point representations across different patches, our method combines the consolidated point representations $\{\tilde{\mathbf{v}}_{i,\mathbf{x}}\}_{i=1}^{N}$ using two sets of weights analogous to those for patch embeddings $\{\mathbf{z}_{i,\mathbf{x}}^{L}\}_{i=1}^{N}$: the holistic–local attention weights $\{\beta_{i,\mathbf{x}}\}_{i=1}^{N}$ (Eq. (2)) and the linguistic grounding weights $\{\mu_{i,\mathbf{x}}^{(\mathbf{v})}\}_{i=1}^{N}$ (introduced in the next subsection). Here, we reuse $\{\beta_{i,\mathbf{x}}\}_{i=1}^{N}$ because they encode region-level importance that applies equally to both feature pathways for $\mathbf{z}_{i,\mathbf{x}}^{L}$ and $\tilde{\mathbf{v}}_{i,\mathbf{x}}$ (more details are available in Appendix C).

3.6 Consolidated-Point Linguistic Grounding for 3D Geometric Semantics

To further enhance the combining of all consolidated point representations $\{\tilde{\mathbf{v}}_{i,\mathbf{x}}\}_{i=1}^{N}$, we introduce another set of weights $\{\mu_{i,\mathbf{x}}^{(\mathbf{v})}\}_{i=1}^{N}$ to capture the alignment between the fine-grained 3D appearance and the textual description of $\mathbf{x}$, which is computed as follows:

$$\mu_{i,\mathbf{x}}^{(\mathbf{v})} = \max\left(\cos\left(\hat{P}\tilde{\mathbf{v}}_{i,\mathbf{x}},\ M_{\text{text}}(t_{\mathbf{x}})\right),\ 0\right). \tag{6}$$

3.7 Combined Embedding and Final 3D Representation

To obtain an integrated representation of $\mathbf{x}$ based on all output embeddings of Point-BERT and consolidated point representations, we combine the `[CLS]` embedding $\mathbf{z}_{0,\mathbf{x}}^{L}$, the patch embeddings $\{\mathbf{z}_{i,\mathbf{x}}^{L}\}_{i=1}^{N}$, and consolidated point representations $\{\tilde{\mathbf{v}}_{i,\mathbf{x}}\}_{i=1}^{N}$ using all weighting factors $\beta_{i,\mathbf{x}}$, $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$, and $\mu_{i,\mathbf{x}}^{(\mathbf{v})}$. Specifically, $\mathbf{r}_{0,\mathbf{x}} := \beta_{0,\mathbf{x}}\mu_{0,\mathbf{x}}^{(\mathbf{z})}\mathbf{z}_{0,\mathbf{x}}^{L}$ and $\mathbf{r}_{i,\mathbf{x}} := \beta_{i,\mathbf{x}}\left(\mu_{i,\mathbf{x}}^{(\mathbf{z})}\mathbf{z}_{i,\mathbf{x}}^{L} + \mu_{i,\mathbf{x}}^{(\mathbf{v})}\tilde{\mathbf{v}}_{i,\mathbf{x}}\right)$ for $i = 1, \ldots, N$. We first aggregate $\{\mathbf{r}_{i,\mathbf{x}}\}_{i=1}^{N}$ by a lightweight Graph Neural Network (GNN), which preserves their 3D spatial structure (more details are available in Appendix D) and produces a single fused patch-level embedding, denoted as $g^{\text{GNN}}(\{\mathbf{r}_{i,\mathbf{x}}\}_{i=1}^{N})$. The output of this GNN is then concatenated with the weighted `[CLS]` embedding $\mathbf{r}_{0,\mathbf{x}}$ to yield the final combined representation $\mathbf{z}_{\mathbf{x}}^{\text{3D}}$ of sample $\mathbf{x}$:

$$\mathbf{z}_{\mathbf{x}}^{\text{3D}} = [\mathbf{r}_{0,\mathbf{x}}; g^{\text{GNN}}(\{\mathbf{r}_{i,\mathbf{x}}\}_{i=1}^{N})] \in \mathbb{R}^{2d}. \tag{7}$$

3.8 Final Loss for Model Training

The final 3D representation in the joint 3D–text embedding space is obtained as $M_{\text{3D}}(\mathbf{x}) = P\mathbf{z}_{\mathbf{x}}^{\text{3D}}$, and the alignment between it and the text embedding of $\mathbf{x}$ is measured as $\Omega(\mathbf{x}, t_{\mathbf{x}}) = \cos\left(M_{\text{3D}}(\mathbf{x}),\ M_{\text{text}}(t_{\mathbf{x}})\right)$. To optimize the MLP g^{MLP}, GNN g^{GNN}, and shared learnable projection layer $\hat{P}$ in our framework using the normal samples, we introduce three loss components.

Category-masked contrastive loss. Let $y^{(b)}$ denote the category label of the b-th sample in a mini-batch of B samples. Since text prompts are deterministic per category, samples with $y^{(b)} = y^{(k)}$

share identical text embeddings, which inflate the standard ULIP-2 loss with duplicate same-category positives (3D-to-text loss $\mathcal{L}_{\text{3D}\to\text{text}}$) and false same-category negatives (text-to-3D loss $\mathcal{L}_{\text{text}\to\text{3D}}$). Let $\mathcal{V}(b) = \{k : k = b \text{ or } y^{(k)} \neq y^{(b)}\}$ and we compute $\mathcal{L}_{\text{3D}\to\text{text}}$ and $\mathcal{L}_{\text{text}\to\text{3D}}$ using $\Omega(\mathbf{x}, t_{\mathbf{x}})$ as the cosine similarity:

$$\begin{aligned}\mathcal{L}_{\text{3D}\to\text{text}} &= -\frac{1}{B}\sum_{b=1}^{B}\log\frac{\exp\big(\Omega(\mathbf{x}^{(b)}, t^{(b)})/\tau\big)}{\sum_{k\in\mathcal{V}(b)}\exp\big(\Omega(\mathbf{x}^{(b)}, t^{(k)})/\tau\big)},\\ \mathcal{L}_{\text{text}\to\text{3D}} &= -\frac{1}{B}\sum_{b=1}^{B}\log\frac{\exp\big(\Omega(\mathbf{x}^{(b)}, t^{(b)})/\tau\big)}{\sum_{k\in\mathcal{V}(b)}\exp\big(\Omega(\mathbf{x}^{(k)}, t^{(b)})/\tau\big)},\end{aligned} \tag{8}$$

where τ is a learnable temperature parameter.

Point-to-patch contrastive loss. We encourage each projected point representation to be closer to its own patch embedding than to other patches' embeddings:

$$\mathcal{L}_{\text{point-patch}} = -\frac{1}{BNK}\sum_{b=1}^{B}\sum_{i=1}^{N}\sum_{j=1}^{K}\log\frac{\exp\Big(\cos(\hat{P}g^{\text{MLP}}(\mathbf{v}_{j,i,\mathbf{x}}^{(b)}),\ \hat{P}\mathbf{z}_{i,\mathbf{x}}^{L,(b)})/\tau\Big)}{\sum_{k=1}^{N}\exp\Big(\cos(\hat{P}g^{\text{MLP}}(\mathbf{v}_{j,i,\mathbf{x}}^{(b)}),\ \hat{P}\mathbf{z}_{k,\mathbf{x}}^{L,(b)})/\tau\Big)}. \tag{9}$$

Intra-patch consistency loss. We regularize the within-patch representation distribution to ensure tight clustering of normal point features:

$$\mathcal{L}_{\text{intra}} = \frac{1}{BN}\sum_{b=1}^{B}\sum_{i=1}^{N}\frac{1}{K(K-1)}\sum_{1\le j\neq k\le K}\Big(1-\cos\Big(\hat{P}g^{\text{MLP}}(\mathbf{v}_{j,i,\mathbf{x}}^{(b)}),\ \hat{P}g^{\text{MLP}}(\mathbf{v}_{k,i,\mathbf{x}}^{(b)})\Big)\Big). \tag{10}$$

Beyond regularization, $\mathcal{L}_{\text{intra}}$ contracts within-patch point-level anomaly score dispersion. Theorem 1 in Sec. 4 will show that, as long as the anomaly margin is preserved, this exponentially improves the probability that an anomalous point attains the largest local deviation score within its patch.

Total loss. The final training objective is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{3D}\to\text{text}} + \mathcal{L}_{\text{text}\to\text{3D}} + \mathcal{L}_{\text{point-patch}} + \lambda_{\text{intra}}\mathcal{L}_{\text{intra}}$, where $\lambda_{\text{intra}} = 0.1$. The three contrastive losses use the same InfoNCE [35] form and temperature. Their magnitudes are comparable in practice, so we use unit weights for them and only down-weight $\mathcal{L}_{\text{intra}}$.

3.9 Inference Procedure

Object-level anomaly score. Following the standard evaluation protocol, in which the object category is provided at test time (most existing methods either assume this explicitly or encode it implicitly through per-category models; Appendix A.4), a test sample $\mathbf{x}_{\text{test}}$ of category c is given, and the object-level anomaly score is:

$$s_{\text{obj}}(\mathbf{x}_{\text{test}}) = -\Omega(\mathbf{x}_{\text{test}}, t_c) = -\cos\left(M_{\text{3D}}(\mathbf{x}_{\text{test}}),\ M_{\text{text}}(t_c)\right). \tag{11}$$

The point-level score below needs no category; a label-free variant is evaluated in Appendix G.7.

Point-level anomaly score. For point j in patch i of point cloud $\mathbf{x}_{\text{test}}$, we compute two complementary scores. The first one is the *within-patch deviation* score, which measures how much a point deviates from the local distribution within its patch:

$$s_j^{\text{local}}(\mathbf{x}_{\text{test}}) = -\cos\Big(\hat{P}g^{\text{MLP}}(\mathbf{v}_{j,i,\mathbf{x}_{\text{test}}}), \bar{\mathbf{v}}_{i,\mathbf{x}_{\text{test}}}\Big), \tag{12}$$

where $\bar{\mathbf{v}}_{i,\mathbf{x}_{\text{test}}} = \frac{1}{K}\sum_{k=1}^{K}\hat{P}g^{\text{MLP}}(\mathbf{v}_{k,i,\mathbf{x}_{\text{test}}})/\|\hat{P}g^{\text{MLP}}(\mathbf{v}_{k,i,\mathbf{x}_{\text{test}}})\|_2$ is the within-patch centroid of the ℓ_2-normalized projected point representations. Theorem 1 and Corollary 1 will motivate this score: tighter within-patch clustering of normal features (encouraged by $\mathcal{L}_{\text{intra}}$) makes anomalous points more likely to attain the highest deviation score within their patch and lowers the anomaly separation required to do so at a target detectability.

The second one is the *point-patch consistency* score, which measures how consistent each point is with its patch embedding from the 3D encoder:

$$s_j^{\text{consist}}(\mathbf{x}_{\text{test}}) = -\cos\Big(\hat{P}g^{\text{MLP}}(\mathbf{v}_{j,i,\mathbf{x}_{\text{test}}}), \hat{P}\mathbf{z}_{i,\mathbf{x}_{\text{test}}}^{L}\Big). \tag{13}$$

The combined point-level anomaly score is $s_j^{\text{point}}(\mathbf{x}_{\text{test}}) = s_j^{\text{local}}(\mathbf{x}_{\text{test}}) + s_j^{\text{consist}}(\mathbf{x}_{\text{test}})$, where both components share the same cosine scale and require no balancing hyperparameter.

4 Theoretical Analysis of Within-Patch Consistency

Among the components of our framework, the intra-patch consistency loss $\mathcal{L}_{\text{intra}}$ (Eq. (10)) and the within-patch deviation score s_j^{local} (Eq. (12)) jointly play a central role in enabling precise point-level anomaly localization. To understand how the former improves the latter, we formally analyze how $\mathcal{L}_{\text{intra}}$ shapes the within-patch score distribution. Specifically, we show that, provided the anomaly margin is preserved and non-anomalous scores satisfy an upper-tail concentration condition, reducing the within-patch dispersion of non-anomalous point scores, which connects to $\mathcal{L}_{\text{intra}}$, *exponentially* increases the probability that an anomalous point attains the highest deviation score in its patch. The analysis concerns only this within-patch component of the point-level score and serves as design guidance for $\mathcal{L}_{\text{intra}}$, e.g., for the trade-off controlled by λ_{intra}, rather than as a performance guarantee for the full framework.

Consider a single patch in which point j^* is anomalous and the remaining $\mathcal{N} = \{1, \dots, K\} \setminus \{j^*\}$ are non-anomalous. For each point j, let s_j denote its anomaly score, defined as the cosine distance to the within-patch centroid (a monotone transformation of s_j^{local} in Eq. (12) that preserves all rankings and thus AUROC). Let $\bar{s}_{\mathcal{N}} = \frac{1}{|\mathcal{N}|}\sum_{j\in\mathcal{N}} s_j$, $\Delta = s_{j^*} - \bar{s}_{\mathcal{N}}$, and $D(j^*) = \Pr(s_{j^*} > \max_{j\in\mathcal{N}} s_j)$ denote the mean non-anomalous-score, anomaly separation, and rank-based detectability, respectively. Let $\sigma^2_{\mathcal{N},\text{tail}}$ denote an upper-tail concentration proxy for non-anomalous-score deviations. Concretely, for a uniformly sampled non-anomalous point $J \sim \text{Unif}(\mathcal{N})$, we consider $\Pr(s_J - \bar{s}_{\mathcal{N}} \geq t) \leq \exp\big(-t^2/(2\sigma^2_{\mathcal{N},\text{tail}})\big)$, $t > 0$. Boundedness gives a loose fallback [5], while $\mathcal{L}_{\text{intra}}$ contracts empirical within-patch score dispersion (Appendix E.1) and the empirical fitted tail scale $\widehat{\sigma}_{\mathcal{N},\text{tail}}$ is much smaller than this worst-case proxy and decreases monotonically with λ_{intra} (Appendix E.2). With these definitions, we formalize how $\mathcal{L}_{\text{intra}}$ translates into rank-based detectability in the theorem below.

Theorem 1 (Within-Patch Consistency Improves Rank-Based Detectability)**.** *For a failure level $\eta \in [0, 1)$, suppose there exists a deterministic margin $\gamma_\eta > 0$ such that $\Pr(\Delta \geq \gamma_\eta) \geq 1 - \eta$. If the non-anomalous-score deviations satisfy the upper-tail concentration condition with proxy $\sigma^2_{\mathcal{N},\text{tail}}$:*

$$D(j^*) \geq 1 - \eta - (K-1)\exp\left(-\frac{\gamma_\eta^2}{2\sigma^2_{\mathcal{N},\text{tail}}}\right). \tag{14}$$

Proof. The proof is available in Appendix E.1.

Insights of Theorem 1. Theorem 1 offers principled insights into the design and understanding of point-level anomaly scoring: the detectability depends *exponentially* on the margin-to-dispersion ratio $\gamma_\eta^2/\sigma^2_{\mathcal{N},\text{tail}}$. Thus, reducing the dispersion of non-anomalous-scores significantly improves the probability of ranking the anomalous point highest, as long as the anomaly margin does not collapse. This predicted exponential dependence is well-aligned with the empirical visualization in Appendix E.2. Additional results in Appendix G.4 report the empirical signal-to-noise ratio $\Delta/\widehat{\sigma}_{\mathcal{N},\text{std}}$, used as an empirical proxy for the margin-to-dispersion ratio in Theorem 1. The results show the predicted trade-off: increasing λ_{intra} reduces $\widehat{\sigma}_{\mathcal{N},\text{std}}$, while over intra-patch regularization can also shrink Δ.

Corollary 1 (Sufficient Condition and Variance–Separation Trade-off)**.** *For any $\delta \in (\eta, 1)$ and $K \geq 2$, a sufficient condition for $D(j^*) \geq 1 - \delta$ is $\sigma^2_{\mathcal{N},\text{tail}} \leq \frac{\gamma_\eta^2}{2\log((K-1)/(\delta-\eta))}$. Consequently, shrinking $\sigma^2_{\mathcal{N},\text{tail}}$ by a factor $\alpha \in (0, 1)$ relaxes the required margin from γ_η to $\sqrt{\alpha}\,\gamma_\eta$.*

Proof. The proof is available in Appendix E.1.

5 Experiments

5.1 Experimental Setup

Datasets and Metrics. We follow popular benchmarks for 3D anomaly detection and use Real3D-AD [23] and Anomaly-ShapeNet [20] to evaluate our method. For metrics, we follow [45] and use the Area Under the Receiver Operating Characteristic Curve (AUROC) for both object-level (**O-AUROC**, ↑) and point-level (**P-AUROC**, ↑) to evaluate the 3D anomaly detection performance, which are both reported in average and Standard Error of the Mean (SEM) values based on 10 random seeds.

Models and Baselines. We adopt ULIP-2 [42] (Point-BERT setup) for the 3D VLM. Our method is evaluated against comprehensive 3D anomaly detection approaches, including BTF [15], M3DM [40], PatchCore [33], CPMF [6], Reg3D-AD [23], Group3AD [51], IMRNet [20], R3D-AD [50], PointAD [49], ISMP [22], PO3AD [43], RPRIL [45], PASDF [47], CASL [46], MC3D-AD [8], PLANE [38], and BTP [18]. Performance metrics for all competing methods are obtained from their publications or publicly available implementations following popular benchmark protocol [45]. Details of implementation are available in Appendix F.

Table 1: AUROC comparison (averaged across all classes, ↑) on Real3D-AD and Anomaly-ShapeNet datasets. **Bold** indicates best, underline indicates second best. "–" indicates results not available under a directly comparable setting. Methods marked with † use a zero-shot protocol (no training data from the target categories). Detailed results are provided in Appendix G.1.

Dataset	Metric	BTF [15]	M3DM [40]	PatchCore [33]	CPMF [6]	Reg3D-AD [23]	Group3AD [51]	IMRNet [20]	R3D-AD [50]	PointAD† [49]	ISMP [22]	PO3AD [43]	RPRIL [45]	PASDF [47]	CASL [46]	MC3D-AD [8]	PLANE [38]	BTP† [18]	Ours
Real3D-AD	O-AUROC	63.5	55.2	59.3	58.6	70.4	75.1	72.5	73.4	74.8	75.7	76.6	78.1	80.2	82.3	78.2	77.7	61.4	**93.1**$_{\pm 0.2}$
	P-AUROC	57.1	63.1	68.2	75.9	70.5	73.5	–	–	73.5	83.6	–	87.9	74.5	88.2	76.8	73.6	84.5	**94.7**$_{\pm 0.2}$
Anomaly-ShapeNet	O-AUROC	49.3	55.4	56.8	55.9	57.2	–	65.9	74.9	–	–	83.9	86.1	90.7	88.7	84.2	83.6	65.2	**95.0**$_{\pm 0.2}$
	P-AUROC	55.0	61.0	58.0	57.3	66.8	–	65.0	–	–	68.9	89.9	88.2	89.6	89.9	74.8	82.1	87.3	**95.6**$_{\pm 0.3}$

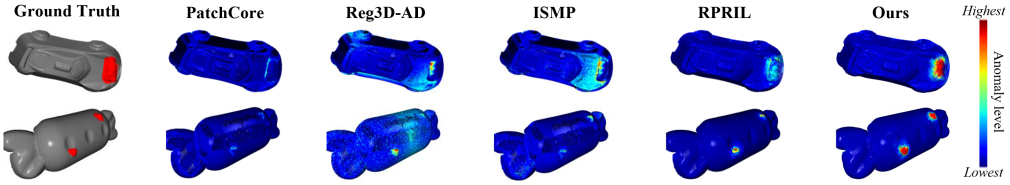


Figure 3: Visualization comparison of anomaly maps on Real3D-AD. Results of competing methods are from [45]. More visualization results are provided in Appendix G.1.

5.2 Main Results

We present the main quantitative and qualitative results in Table 1 and Figure 3, respectively. Additional results are provided in the appendix, including detailed per-category quantitative and qualitative results and a comparison with native 3D-VLM methods (Appendix G.1), visualization of our proposed weights (Appendix G.2), and efficiency and scalability analysis (Appendix G.3).

Quantitative Results. Table 1 summarizes the average object- and point-level AUROC on both benchmarks. On Real3D-AD, our method achieves 93.1% O-AUROC and 94.7% P-AUROC, substantially and consistently surpassing the second-best method by **10.8** and **6.5** points, respectively. On Anomaly-ShapeNet, our method attains 95.0% O-AUROC and 95.6% P-AUROC, consistently and meaningfully outperforming the runner-up by **4.3** and **5.7** points. Notably, our method achieves the best average AUROC in all four settings. Per category, it is best or tied-best in 6/12 (O-AUROC) and 10/12 (P-AUROC) categories on Real3D-AD and in 22/40 and 21/40 on Anomaly-ShapeNet; overall, it is best or second-best in 95 of the 104 category–metric pairs and third in the other nine (Appendix G.1).

Qualitative Results. We show a comparison of anomaly map visualizations in Figure 3. As can be observed, our method generates more accurate anomaly maps on Real3D-AD, which leads to a significant improvement in P-AUROC compared to competing methods.

Table 2: Results of ablation study for object-level anomaly detection (averaged AUROC across all classes, ↑) on Real3D-AD and Anomaly-ShapeNet datasets. Each column shows performance when removing one component. The last column shows our method equipped with all mechanisms.

Dataset	Combining weights				Retain point representations	Loss function		Combiner	**Full**
	W/o β	W/o $\mu^{(\mathbf{z})}$	W/o $\mu^{(\mathbf{v})}$	W/o w	W/o entire $\mathbf{v}$ branch	W/o $\mathcal{L}_{\text{point-patch}}$	W/o $\mathcal{L}_{\text{intra}}$	W/o GNN	**Ours**
Real3D-AD	89.4	91.2	92.3	91.9	86.0	92.1	92.6	90.6	**93.1**
Anomaly-ShapeNet	90.3	92.6	94.0	93.6	88.8	93.7	94.5	91.2	**95.0**

Table 3: Results of ablation study for point-level anomaly detection (averaged AUROC across all classes, ↑) on Real3D-AD and Anomaly-ShapeNet. Other conventions are the same as in Table 2.

Dataset	Combining weights				Anomaly score		Loss function		Combiner	**Full**
	W/o β	W/o $\mu^{(\mathbf{z})}$	W/o $\mu^{(\mathbf{v})}$	W/o w	W/o s^{local}	W/o s^{consist}	W/o $\mathcal{L}_{\text{point-patch}}$	W/o $\mathcal{L}_{\text{intra}}$	W/o GNN	**Ours**
Real3D-AD	94.0	94.2	94.4	94.3	90.3	91.8	90.0	87.9	93.8	**94.7**
Anomaly-ShapeNet	94.7	95.1	95.2	95.1	92.0	92.5	89.3	90.4	94.5	**95.6**

5.3 Ablation Studies

We present the results of ablation studies for object- and point-level anomaly detection in Tables 2 and 3, respectively. Sensitivity analyses of $\mathcal{L}_{\text{intra}}$ and of the point-level anomaly score, robustness to noisy point clouds, controlled ablations, and mechanism diagnostics (including defect-size sensitivity) are available in Appendices G.4, G.5, G.6, G.7, and G.8, respectively.

Our Proposed Four Combining Weights. Among the four weighting mechanisms proposed by us, β has the largest object-level impact, confirming that holistic–local attention (β) is crucial for anomaly detection. All four weights (β, $\mu^{(\mathbf{z})}$, $\mu^{(\mathbf{v})}$, and w) show comparatively small point-level drops, as they do not directly modulate the point-level scoring.

Point Representation Branch and Anomaly Score. Removing the entire $\mathbf{v}$-branch causes the largest drop at the object-level, revealing that retention of native pre-max-pooling point representations from the tokenizer of the 3D encoder is critical; linear probes confirm that these representations encode surface normals and curvature (Appendix G.8). At the point level, both anomaly score components contribute: s^{local} has a larger impact than s^{consist}, and neither alone matches the full model.

Loss Function and Combiner. Both loss terms are critical for point-level performance and show a dataset-dependent pattern: $\mathcal{L}_{\text{intra}}$ has a larger impact on Real3D-AD, while $\mathcal{L}_{\text{point-patch}}$ dominates on Anomaly-ShapeNet. At the object level, both losses yield modest performance improvements through a better-trained MLP. As for the Combiner, replacing the GNN with a mean pooling reduces object-level performance, indicating that patch structural information is critical for anomaly detection.

Backbone and Language. A direct ULIP-2 baseline with the same frozen backbone, prompts, and protocol but none of our modules reaches only 53.8/54.4 (Real3D-AD) and 55.2/55.7 (Anomaly-ShapeNet) O-/P-AUROC, so the gains come from the FACETS modules and losses. Replacing the category text embeddings with learnable category embeddings costs 1.0–2.0 points and removing language costs 2.0–5.5 points: category-level 3D–text grounding gives a meaningful but incremental gain over learnable category conditioning, while other prompt templates or a prompt ensemble change results by at most 0.7 points (Appendix G.7).

6 Conclusions

We present FACETS, a framework that leverages a 3D vision–language model for 3D anomaly detection, moving beyond existing reconstruction- and memory-bank-based methods. Our framework fully exploits the representational capacity of the 3D VLM via four cross-granularity mechanisms: holistic–local attention for fine-grained geometric details modeling, patch linguistic grounding to enrich 3D–textual semantics, and point geometric retention and consolidated-point linguistic grounding for subtle point-level geometric cues recovery and semantics refinement. Crucially, FACETS explicitly leverages native point features to provide the substrate for the entire framework. Mathematical analysis of the within-patch component of our point-level anomaly score delivers concrete insights into its design. Extensive experiments show that FACETS achieves the best average AUROC across datasets and metrics, consistently and substantially surpassing existing methods.

FACETS inherits the representational coverage of the frozen ULIP-2 backbone, uses simple category-level prompt templates, and assumes that the test category is known for standard object-level inference; open-world generalization to unseen categories remains future work. Appendix H discusses these limitations and possible extensions.

Acknowledgments and Disclosure of Funding

This research has been funded by the Industrial Technology Innovation Program [P0030255, "Project OptiBatt - AI: Digital Twin-Integrated AI Agent for Manufacturing of EV Battery Systems – Cooling, High Voltage, and ICB Modules" led by Sungwoo Hitech Co., Ltd.] of the Ministry of Trade, Industry and Energy of the Republic of Korea. This work was also supported by the Digital Research Alliance of Canada through the Fast Track (FT) 3745 [RRG 2026]. The authors declare no competing interests.

References

[1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9592–9600, 2019. doi: 10.1109/CVPR.2019.00982.

[2] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The MVTec 3D-AD dataset for unsupervised 3D anomaly detection and localization. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP) – Volume 5: VISAPP*, pages 202–213. SciTePress, 2022. doi: 10.5220/0010865000003124.

[3] Paul J. Besl and Neil D. McKay. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–606. SPIE, 1992. doi: 10.1117/12.57955.

[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 (ACCV)*, volume 13845 of *Lecture Notes in Computer Science*, pages 459–475. Springer, 2023. doi: 10.1007/978-3-031-26348-4_27.

[5] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. doi: 10.1093/acprof:oso/9780199535255.001.0001.

[6] Yunkang Cao, Xiaohao Xu, and Weiming Shen. Complementary pseudo multimodal feature for point cloud anomaly detection. *Pattern Recognition*, 156:110761, 2024. doi: 10.1016/j.patcog.2024.110761.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021. doi: 10.1109/ICCV48922.2021.00951.

[8] Jiayi Cheng, Can Gao, Jie Zhou, Jiajun Wen, Tao Dai, and Jinbao Wang. MC3D-AD: A unified geometry-aware reconstruction model for multi-category 3D anomaly detection. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 837–845. International Joint Conferences on Artificial Intelligence Organization, 2025. doi: 10.24963/ijcai.2025/94.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[10] Juan Du, Chengyu Tao, Xuanming Cao, and Fugee Tsung. 3D vision-based anomaly detection in manufacturing: A survey. *Frontiers of Engineering Management*, 12(2):343–360, 2025. doi: 10.1007/s42524-025-4189-9.

[11] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 70, pages 1263–1272. PMLR, 2017.

[12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 249–256. PMLR, 2010.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. doi: 10.1109/ICCV.2015.123.

[14] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

[15] 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 Workshops (CVPRW)*, pages 2968–2977, 2023. doi: 10.1109/CVPRW59228.2023.00298.

[16] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. WinCLIP: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19606–19616, 2023. doi: 10.1109/CVPR52729.2023.01878.

[17] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955. doi: 10.1002/nav.3800020109.

[18] Kaiqiang Li, Gang Li, Mingle Zhou, Min Li, Delong Han, and Jin Wan. Back to point: Exploring point-language models for zero-shot 3D anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14167–14177, 2026.

[19] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, pages 3538–3545, 2018. doi: 10.1609/aaai.v32i1.11604.

[20] Wenqiao Li, Xiaohao Xu, Yao Gu, Bozhong Zheng, Shenghua Gao, and Yingna Wu. Towards scalable 3D anomaly detection and localization: A benchmark via 3D anomaly synthesis and a self-supervised learning network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22207–22216, 2024. doi: 10.1109/CVPR52733.2024.02096.

[21] Yiting Li, Xulei Yang, Jing Zhang, Sichao Tian, Jingyi Liao, and Fayao Liu. PIRN: Prototypical-based intra-modal reconstruction with normality communication for multi-modal anomaly detection. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

[22] 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 (AAAI)*, volume 39, pages 5146–5154, 2025. doi: 10.1609/aaai.v39i5.32546.

[23] 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. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 36, pages 30402–30415, 2023. doi: 10.52202/075280-1324.

[24] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[26] Meta Research. fvcore. `https://github.com/facebookresearch/fvcore`, 2021.

[27] Atsuyuki Miyai, Qing Yu, Go Irie, and Kiyoharu Aizawa. LoCoOp: Few-shot out-of-distribution detection via prompt learning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 76298–76310, 2023. doi: 10.52202/075280-3335.

[28] Karl Pearson. LIII. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. doi: 10.1080/14786440109462720.

[29] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017. doi: 10.1109/CVPR.2017.16.

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 5099–5108, 2017.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 139, pages 8748–8763. PMLR, 2021.

[32] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 12116–12128, 2021.

[33] 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 (CVPR)*, pages 14318–14328, 2022. doi: 10.1109/CVPR52688.2022.01392.

[34] Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. doi: 10.2307/1412159.

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. doi: 10.48550/arXiv.1807.03748.

[36] Chengjie Wang, Wenbing Zhu, Bin-Bin Gao, Zhenye Gan, Jiangning Zhang, Zhihao Gu, Shuguang Qian, Mingang Chen, and Lizhuang Ma. Real-IAD: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22883–22892, 2024. doi: 10.1109/CVPR52733.2024.02159.

[37] Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. CLIPN for zero-shot OOD detection: Teaching CLIP to say no. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1802–1812, 2023. doi: 10.1109/ICCV51070.2023.00173.

[38] Jiaxiang Wang, Haote Xu, Xiaolu Chen, Haodi Xu, Yue Huang, Xinghao Ding, and Xiaotong Tu. Exploiting point-language models with dual-prompts for 3D anomaly detection. *Expert Systems with Applications*, 298:129758, 2026. doi: 10.1016/j.eswa.2025.129758.

[39] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019. doi: 10.1145/3326362.

[40] 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 (CVPR)*, pages 8032–8041, 2023. doi: 10.1109/CVPR52729.2023.00776.

[41] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP: Learning a unified representation of language, images, and point clouds for 3D understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1189, 2023. doi: 10.1109/CVPR52729.2023.00120.

[42] Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP-2: Towards scalable multimodal pre-training for 3D understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27091–27101, 2024. doi: 10.1109/CVPR52733.2024.02558.

[43] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1353–1362, 2025. doi: 10.1109/CVPR52734.2025.00134.

[44] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-training 3D point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19313–19322, 2022. doi: 10.1109/CVPR52688.2022.01871.

[45] 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. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pages 5806–5832, 2025. doi: 10.52202/085713-0180.

[46] Yaohua Zha, Xue Yuerong, Chunlin Fan, Yuansong Wang, Tao Dai, Ke Chen, and Shu-Tao Xia. CASL: Curvature-augmented self-supervised learning for 3D anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pages 12340–12348, 2026. doi: 10.1609/aaai.v40i15.38226.

[47] 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 (ICCV)*, pages 27063–27072, 2025. doi: 10.1109/ICCV51701.2025.02512.

[48] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[49] Qihang Zhou, Jiangtao Yan, Shibo He, Wenchao Meng, and Jiming Chen. PointAD: Comprehending 3D anomalies from points and pixels for zero-shot 3D anomaly detection. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 84866–84896, 2024. doi: 10.52202/079017-2695.

[50] Zheyuan Zhou, Le Wang, Naiyu Fang, Zili Wang, Lemiao Qiu, and Shuyou Zhang. R3D-AD: Reconstruction via diffusion for 3D anomaly detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 15094 of *Lecture Notes in Computer Science*, pages 91–107. Springer, 2024. doi: 10.1007/978-3-031-72764-1_6.

[51] 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 (ACM MM)*, pages 4680–4689. ACM, 2024. doi: 10.1145/3664647.3680919.

Supplementary Material

Table of Contents

A Extended Related Work 16
A.1 Reconstruction-Based Methods 16
A.2 Memory-Bank-Based Methods 16
A.3 Vision–Language Models for 3D Understanding 17
A.4 Training Regime and Category Usage of Compared Methods 18

B Detailed Architecture Diagram and Design Rationale 18
B.1 Design Rationale 20

C Design of the Shared $\beta_{i,\mathbf{x}}$ for $\mathbf{z}^L_{i,\mathbf{x}}$ and $\tilde{\mathbf{v}}_{i,\mathbf{x}}$ 20

D Details of GNN in the Combiner 24

E Theoretical and Empirical Analysis of Within-Patch Consistency 24
E.1 Proof of Theorem 1 and Corollary 1 24
E.2 Empirical Illustration of Theorem 1 28

F Details of Implementation 30

G Supplementary Results 32
G.1 Extended Main Results 32
G.2 Visualization of Our Weights $\beta_{i,\mathbf{x}}$, $\mu^{(\mathbf{z})}_{i,\mathbf{x}}$, and $\mu^{(\mathbf{v})}_{i,\mathbf{x}}$ 32
G.3 Efficiency and Scalability Analysis 36
G.4 Sensitivity Analysis for λ_{intra} 37
G.5 Sensitivity Analysis for Our Point-level Anomaly Score 38
G.6 Robustness to Noisy Point Clouds 40
G.7 Controlled Ablations 41
G.8 Mechanism Diagnostics 42

H Limitations and Future Work 42

A Extended Related Work

This section provides an extended discussion of existing methods for 3D anomaly detection in addition to the concise overview in Sec. 2.

A.1 Reconstruction-Based Methods

Reconstruction-based methods learn to faithfully reconstruct normal samples. Thus, anomalies are detected where reconstruction errors are large, based on the premise that a model trained exclusively on normal samples will fail to reconstruct anomalous regions.

IMRNet [20] introduces a self-supervised learning network for scalable anomaly detection and localization, which learns to reconstruct normal samples from synthetically corrupted inputs.

R3D-AD [50] leverages diffusion models for reconstruction, which learn to iteratively denoise corrupted samples back to their clean normal forms. Anomaly scores are derived from the discrepancy between input and reconstructed point clouds, exploiting the diffusion model's learned prior over normal geometry.

PO3AD [43] addresses a limitation of prior reconstruction methods that distribute attention equally across normal and pseudo-anomalous regions. The authors observe that equal weighting dilutes the model's focus on actual anomalous deviations, and propose to emphasize learning point offsets specifically targeting more informative pseudo-abnormal points. PO3AD additionally introduces a surface-normal-guided augmentation technique to create credible pseudo-anomalies that enhance training.

PASDF [47] takes a fundamentally different approach by leveraging Signed Distance Functions (SDFs) for continuous geometric representation. The method introduces a Pose-Aware SDF framework that explicitly disentangles pose and shape through a pose-wise alignment module, which enables robust registration of input samples to a learned canonical coordinate frame. By learning a canonicalized SDF representation, PASDF captures global geometric context and detects subtle anomalies that local feature methods might miss. The method additionally generates high-fidelity repair templates, unifying anomaly detection and repair in a single framework.

CASL [46] proposes a curvature-augmented self-supervised learning framework built on a reconstruction paradigm. A key finding is that point curvature alone, used as an anomaly score, already outperforms several classical self-supervised and dedicated anomaly detection models. Built on a U-Net architecture, CASL introduces multi-scale curvature prompts to guide coordinate prediction, achieving leading detection performance on Real3D-AD and Anomaly-ShapeNet without relying on task-specific anomaly detection mechanisms.

MC3D-AD [8] proposes a unified geometry-aware reconstruction model for multi-category 3D anomaly detection, so that a single model serves all categories. At inference, each test sample is registered to a template of its category, which dominates its runtime (Appendix G.3). Its published results are included in Tables 1, 8, and 9.

PIRN [21] proposes a prototype-driven reconstruction framework for multimodal anomaly detection using both RGB and 3D modalities. PIRN introduces three key components: Balanced Prototype Assignment (BPA) via optimal transport to ensure uniform prototype utilization and prevent codebook collapse, Adaptive Prototype Refinement (APR) with gated GRU updates to dynamically expand coverage to unseen normal variations, and Multimodal Normality Communication (MNC) that exchanges cross-modal normality cues via gated cross-attention. The method is validated on earlier benchmarks such as MVTec 3D-AD [2], Eyecandies [4], and Real-IAD [36], achieving state-of-the-art performance especially in few-shot scenarios. However, the method is not evaluated on the more recent and popular Real3D-AD [23] and Anomaly-ShapeNet [20] benchmarks that are the focus of this work.

A.2 Memory-Bank-Based Methods

Memory-bank-based methods store representative features extracted from normal training samples and detect anomalies by measuring the distance between test sample features and the stored normal

feature distribution. These methods typically leverage pre-trained feature extractors and do not require training a reconstruction network.

BTF [15] adapts classical 3D features, which are hand-crafted descriptors such as Fast Point Feature Histograms (FPFH) for anomaly detection via a nearest-neighbor memory bank. Despite its simplicity, BTF demonstrates that classical 3D features remain competitive baselines, establishing a strong foundation for subsequent memory-bank methods.

PatchCore [33] constructs a coreset-based memory bank from locally aware patch features extracted by a pre-trained network. Although originally proposed for 2D anomaly detection, PatchCore has been widely adapted for 3D settings as a representative memory-bank baseline, demonstrating the effectiveness of coreset sampling for reducing memory requirements while maintaining detection performance.

M3DM [40] extends the memory-bank paradigm to the multimodal setting by performing hybrid fusion of 2D image features and 3D point cloud features. By combining complementary information from both modalities, M3DM demonstrates improved detection over single-modality methods, though it requires both RGB images and point clouds at inference time.

Reg3D-AD [23] introduces point cloud registration as a preprocessing step for the memory-bank pipeline. Alongside the Real3D-AD dataset, the method demonstrates that registration-based alignment of test samples to training samples significantly improves feature matching quality, addressing pose variation that can confound distance-based anomaly scoring.

CPMF [6] generates complementary pseudo-multimodal features from point clouds alone, enabling multimodal-style anomaly detection without requiring actual 2D images. By constructing complementary feature views of the 3D data, CPMF achieves competitive performance while eliminating the dependence on paired RGB-3D inputs.

Group3AD [51] targets high-resolution point clouds with group-level feature contrastive learning: an Intercluster Uniformity Network spreads the features of different object parts uniformly in the feature space, an Intracluster Alignment Network tightens the features within each part, and an Adaptive Group-Center Selection based on geometric information samples potential anomalous regions more densely at inference. Like the other memory-bank-based methods, it builds a per-category memory bank and relies on purely visual features without textual semantics.

ISMP [22] introduces Internal Spatial Modality Perception, which leverages internal structural information of point clouds rather than focusing solely on external surface features. The method proposes a Spatial Insight Engine (SIE) that abstracts complex internal information into essential global features by projecting point clouds from internal viewpoints. Combined with enhanced keypoint feature extraction and a feature filtering module, ISMP achieves significant improvements in both object-level and point-level anomaly detection.

RPRIL [45] addresses the issues of inconsistent feature transformations and limited discriminative capacity that are common in memory-bank methods, particularly when registration fails. Based on the insight that feature extraction and registration both rely on modeling local geometric structures, RPRIL proposes a registration-induced, rotation-invariant feature extraction framework. By embedding feature extraction into the registration learning process, the framework jointly optimizes alignment and representation learning, yielding features that are both robust to rotations and highly effective for anomaly detection.

A.3 Vision–Language Models for 3D Understanding

Vision–Language Models (VLMs) have revolutionized 2D representation learning through large-scale contrastive pre-training. CLIP [31] aligns image and text embeddings in a shared space, enabling zero-shot and few-shot transfer across diverse visual tasks. Extending this to 3D, ULIP [41] learns a unified representation across images, text, and 3D point clouds by leveraging a pre-trained CLIP model and aligning a 3D encoder with the existing image-text space using automatically synthesized triplets. ULIP-2 [42] further advances this paradigm with a larger 3D backbone (Point-BERT [44]), expanded pre-training data, and fine-grained language descriptions that go beyond simple category labels to fine-grained descriptive captions.

PointAD [49] is a representative 2D-VLM-based method for 3D anomaly detection, building on multiple works that use a VLM for 2D anomaly detection (e.g., WinCLIP [16] and AnomalyCLIP [48]). PointAD relies on a 2D VLM (i.e., CLIP) and operates through multi-view rendering: it projects point clouds into multiple 2D images, computes pixel-level anomaly predictions via CLIP's patch-text similarity, and aggregates predictions across views. While PointAD demonstrates the promise of vision–language representations for 3D anomaly detection, this rendering-based approach has inherent limitations. First, it is not 3D-spatially aware: each point's features are derived from independent 2D projections without correlation to 3D neighbors, losing local geometric context. Second, it requires multiple forward passes through CLIP's image encoder for each rendered view, incurring significant computational overhead. Third, 3D anomalies that are not visible in any of the 2D projections, such as internal geometric defects or subtle surface deviations occluded from all rendered viewpoints, may be missed entirely.

PLANE [38] is a native 3D-VLM-based method that also builds on ULIP-2 under the multi-class-one-model paradigm. It adapts ULIP-2 with learnable dual prompts trained on synthetic pseudo-anomalies and scores anomalies by intermediate patch–text similarities. It neither retains the tokenizer's pre-max-pooling point features nor addresses the geometry-agnostic aggregation that FACETS targets.

BTP [18] explores point–language models (ULIP-2) for zero-shot 3D anomaly detection: it is trained on one labeled auxiliary category and uses no data from the target categories, so its results are not directly comparable to the unsupervised setting of this work. PLANE and BTP are included in Tables 1, 8, and 9, and Table 10 (Appendix G.1) contrasts their protocols and results with FACETS.

Like PLANE and BTP, our proposed FACETS leverages a native 3D VLM for 3D anomaly detection, moving beyond the reconstruction-based and memory-bank-based approaches that have dominated the field; it is distinguished by retaining the native pre-max-pooling point features of the tokenizer and grounding them, together with patch features, to category-level text. By building directly on ULIP-2's 3D encoder (Point-BERT), FACETS operates natively in 3D space, explicitly models local 3D geometric structures, and exploits the category-level alignment between point clouds and text. Furthermore, because ULIP-2's pre-trained embedding space provides a shared semantic scaffold across object categories, FACETS naturally supports multi-category training with a single unified model, eliminating the need for per-category models required by the majority of existing methods.

A.4 Training Regime and Category Usage of Compared Methods

One unified model and category-free inference are distinct properties. Table 4 summarizes how each compared method is trained and whether its inference needs an externally supplied category label. Per-category methods need the label implicitly, since each test sample must be routed to its category-specific model, memory bank, or registration template; unified methods differ in whether they use category-conditioned prompts or templates. FACETS trains one model per dataset: its object-level score uses the prompt t_c of the provided category, while its point-level score is prompt-independent. A label-free variant of FACETS that selects the best-matching seen category is evaluated in Appendix G.7.

B Detailed Architecture Diagram and Design Rationale

Figure 4 provides a detailed diagram of the FACETS framework in the training phase in addition to the high-level overview in Figure 2.

Panel (a) illustrates the full forward pass. An input point cloud $\mathbf{x}$ is first partitioned into N point patches $\{\mathcal{P}_{i,\mathbf{x}}\}_{i=1}^{N}$, each containing K points, and passed through the tokenizer (a Mini-PointNet) to produce initial patch embeddings that are refined by the frozen 3D Encoder into $\{\mathbf{z}_{i,\mathbf{x}}^{L}\}_{i=0}^{N}$. In parallel, the learnable MLP $g^{\text{MLP}}(\cdot)$ projects the native pre-max-pooling point-level features $\{\{\mathbf{v}_{j,i,\mathbf{x}}\}_{j=1}^{K}\}_{i=1}^{N}$ extracted from the middle of the tokenizer into the patch embedding space. Point geometric retention weights $\{\{w_{j,i,\mathbf{x}}\}_{j=1}^{K}\}_{i=1}^{N}$ (cosine similarity between the projected point features and patch output embeddings) aggregate these point features into consolidated point representations $\{\tilde{\mathbf{v}}_{i,\mathbf{x}}\}_{i=1}^{N}$. Meanwhile, the text prompt $t_{\mathbf{x}}$ is encoded by the frozen Text Encoder into $M_{\text{text}}(t_{\mathbf{x}})$, which is used to compute the patch and consolidated-point linguistic grounding weights: $\{\mu_{i,\mathbf{x}}^{(\mathbf{z})}\}_{i=0}^{N}$ for the `[CLS]` and patch output embeddings and $\{\mu_{i,\mathbf{x}}^{(\mathbf{v})}\}_{i=1}^{N}$ for the consolidated point representations

Table 4: Training regime of the compared methods and their use of an externally supplied category label at test time. Methods with the same regime are grouped.

Method(s)	**Training regime**	**Externally supplied category label at test**
BTF [15], M3DM [40], PatchCore [33], CPMF [6], Group3AD [51], ISMP [22]	Per-category memory, coreset, or pseudo-feature bank	Yes, implicit (bank routing)
Reg3D-AD [23], RPRIL [45]	Per-category, registration-based (template or features plus banks)	Yes, implicit (template selection or bank routing)
IMRNet [20], R3D-AD [50], PO3AD [43], CASL [46], PASDF [47]	Per-category reconstruction, diffusion, or canonical SDF	Yes, implicit (model selection)
PointAD [49], BTP [18]	Zero-shot transfer (prompts learned on auxiliary data)	No (object-agnostic normal/abnormal prompts)
PLANE [38]	Unified multi-category	Yes, explicit (class-specific prompts)
MC3D-AD [8]	Unified multi-category	Yes, implicit (category template selection for test-time registration)
FACETS (ours)	Unified multi-category	Yes for the object-level score; no for the point-level score

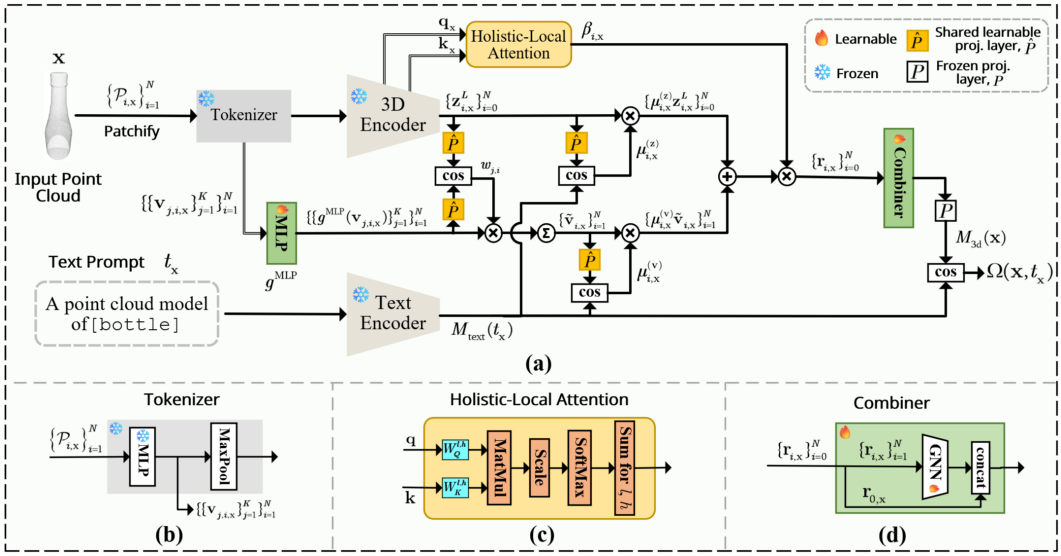


Figure 4: **Detailed Overview of our proposed FACETS framework.** Panel (a) shows the full computation graph; panels (b)–(d) expand the tokenizer, holistic–local attention, and combiner, respectively.

(both via cosine similarity). The holistic–local attention weights $\{\beta_{i,\mathbf{x}}\}_{i=0}^{N}$ are aggregated across all layers and heads of the 3D Encoder. These three sets of weights: $\beta_{i,\mathbf{x}}$, $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$, and $\mu_{i,\mathbf{x}}^{(\mathbf{v})}$, jointly modulate the [CLS] and patch output embeddings, and consolidated point representations, which are then combined into $\{\mathbf{r}_{i,\mathbf{x}}\}_{i=0}^{N}$ and passed to the Combiner to produce the final 3D representation $M_{\text{3D}}(\mathbf{x})$.

Panels (b)–(d) expand three key sub-modules in FACETS. The **Tokenizer** (b) applies a Mini-PointNet, which is composed of an MLP followed by channel-wise max-pooling over the K points in each patch to produce the initial patch embedding. We propose to extract the pre-max-pooling point features, which are native point-level representations. The **Holistic–Local Attention** (c) details the attention mechanism: the [CLS] query $\mathbf{q}$ and patch keys $\mathbf{k}$ are projected by per-layer, per-head matrices $W_Q^{l,h}$ and $W_K^{l,h}$, followed by scaled dot-product attention with softmax normalization, and the resulting coefficients are averaged across all L layers and H heads to produce $\{\beta_{i,\mathbf{x}}\}_{i=0}^{N}$. The **Combiner** (d) takes the weighted [CLS] embedding $\mathbf{r}_{0,\mathbf{x}}$ and the weighted patch-level embeddings $\{\mathbf{r}_{i,\mathbf{x}}\}_{i=1}^{N}$. The weighted patch-level embeddings $\{\mathbf{r}_{i,\mathbf{x}}\}_{i=1}^{N}$ are aggregated by a single-layer GNN, and the resulting fused embedding is concatenated with $\mathbf{r}_{0,\mathbf{x}}$ to form the final combined representation $\mathbf{z}_{\mathbf{x}}^{\text{3D}} \in \mathbb{R}^{2d}$.

B.1 Design Rationale

Table 5 summarizes the rationale of each design choice in FACETS and separates the kinds of support behind it: the structural rationale and its claimed status, mechanism-level or formal support, and empirical evidence. It does not imply that any choice is theoretically derived or uniquely optimal. In brief, the rectified cosine similarity is a simple nonnegative gate; the unnormalized point sum lets the branch magnitude reflect the aggregate parent-aligned point evidence; the shared $\beta_{i,\mathbf{x}}$ gates region importance across both feature paths, the branch-specific $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$ and $\mu_{i,\mathbf{x}}^{(\mathbf{v})}$ gate category alignment, and addition combines contextual patch features with retained local geometry in the same space. The three contrastive losses share the same InfoNCE form, temperature, and comparable magnitudes, which motivates unit weights; only $\mathcal{L}_{\text{intra}}$ is down-weighted and swept (Appendix G.4). The equal-weight sum of the two point-level scores follows from their shared cosine scale; performance peaks at equal weighting and is robust nearby (Appendix G.5).

C Design of the Shared $\beta_{i,\mathbf{x}}$ for $\mathbf{z}_{i,\mathbf{x}}^{L}$ and $\tilde{\mathbf{v}}_{i,\mathbf{x}}$

We first provide a formal justification for why $\beta_{i,\mathbf{x}}$ is a valid importance weight for both the patch output embeddings $\mathbf{z}_{i,\mathbf{x}}^{L}$ and the consolidated point representations $\tilde{\mathbf{v}}_{i,\mathbf{x}}$, and then present a series of comprehensive empirical analyses to investigate it.

Proposition 1 (Region-Level Nature of $\beta_{i,\mathbf{x}}$)**.** *Let $\beta_{i,\mathbf{x}} = \frac{1}{LH}\sum_{l=1}^{L}\sum_{h=1}^{H}\alpha_{i,0}^{l,h}$ be the holistic–local attention weight averaged across all L layers and H heads of Point-BERT, where each attention logit decomposes as:*

$$(\mathbf{z}_i^{l-1}+\mathbf{p}_i)^\top A^{l,h}(\mathbf{z}_0^{l-1}+\mathbf{p}_0) = \underbrace{\mathbf{p}_i^\top A^{l,h}\mathbf{p}_0}_{\textit{pos-pos}} + \underbrace{(\mathbf{z}_i^{l-1})^\top A^{l,h}\,\mathbf{z}_0^{l-1}}_{\textit{content-content}} + \underbrace{\mathbf{p}_i^\top A^{l,h}\,\mathbf{z}_0^{l-1}}_{\textit{pos-content}} + \underbrace{(\mathbf{z}_i^{l-1})^\top A^{l,h}\,\mathbf{p}_0}_{\textit{content-pos}}, \tag{15}$$

with $A^{l,h} = W_K^{l,h}(W_Q^{l,h})^\top$ and positional embedding $\mathbf{p}_i$ derived from the 3D patch center. Then:

1. ***(Geometric consistency across layers.)** Since $\mathbf{p}_i$ is identical across all transformer layers in Point-BERT, the position-dependent terms (i.e., pos-pos and pos-content in Eq. (15)[2]) carry a consistent geometric signal determined by the center coordinates of patch i at every layer. When summed across all L layers during the computation of $\beta_{i,\mathbf{x}}$, these terms accumulate coherently, making $\beta_{i,\mathbf{x}}$ sensitive to the geometric properties of the 3D region for patch i.*

2. ***(Multi-scale semantic aggregation.)** The content-dependent terms vary across layers as representations evolve: early layers capture local geometric structure while later layers encode global semantic relationships [9, 32]. Averaging over all layers and heads yields*

[2]Among the four terms, content-pos depends on patch content but not on the patch positional embedding $\mathbf{p}_i$ and therefore does not provide patch-specific positional information.

Table 5: Design rationale of FACETS: for each design choice, its rationale and claimed status, its mechanism-level or formal support, and its empirical evidence. Component ablations support the individual factors, not the uniqueness of any particular form.

Design choice	**Rationale / claimed status**	**Mechanism / formal support**	**Empirical evidence**
Point geometric retention	Preserve tokenizer features otherwise removed by max-pooling.	Linear probes: retained features encode normals and curvature (Table 16).	Largest object-level drop when removed (Table 2).
w and the unnormalized sum	Gate points by parent-patch alignment while keeping the aggregate aligned evidence; no optimality claim.	Point–patch cosine of the consistency pathway; bounded-scale and invariance argument (Appendix G.7).	W/o w (Tables 2–3); normalized variant (Table 15).
Shared β	Reuse one region saliency for two representations of the same patch.	Proposition 1: analytical motivation, not optimality; Figures 5–7: cross-instance stability.	W/o β (Tables 2–3); shared vs. separate β (Table 15).
$\beta_{i,\mathbf{x}}\big(\mu_{i,\mathbf{x}}^{(\mathbf{z})}\mathbf{z}_{i,\mathbf{x}}^{L} + \mu_{i,\mathbf{x}}^{(\mathbf{v})}\tilde{\mathbf{v}}_{i,\mathbf{x}}\big)$	Conjunctive saliency and category gating, then additive fusion; not uniquely optimal.	Complementary geometric vs. category-alignment patterns (Appendix G.2).	Component ablations support the factors, not the uniqueness of this form (Tables 2–3).
Category-masked 3D–text loss	Exclude duplicate texts (3D-to-text) and false same-category negatives (text-to-3D); the same cosine defines s_{obj}.	Structural: the denominator construction in Eq. (8).	Controlled language ablations assess the grounding pathway overall (Table 15).
$\mathcal{L}_{\text{point-patch}}$ and s_j^{consist}	Use the same point-to-parent-patch cosine at training and test time.	Direct correspondence between the training loss and the test score.	Loss ablation (Table 3); score-component and weight sensitivity (Appendix G.5).
$\mathcal{L}_{\text{intra}}$ and s_j^{local}	Contract the within-patch dispersion of normal scores.	Lemma 1 is deterministic; Theorem 1 and Corollary 1 are conditional guidance for this pair only.	Loss ablation (Table 3); CCDF and dispersion analysis (Appendix E.2); λ_{intra} sweep (Appendix G.4).
GNN combiner	One spatial k-NN layer with relative positions, limiting over-smoothing.	Explicit spatial graph construction (Appendix D).	W/o GNN (Tables 2–3); k_{GNN} sensitivity (Table 6).

an importance measure reflecting a comprehensive multi-scale assessment of each patch region's contribution to the holistic representation.

Together, these properties establish $\beta_{i,\mathbf{x}}$ as a region-level importance weight that characterizes the 3D patch region for patch i, rather than a quality assessment specific to any particular feature representation of that region. Since $\mathbf{z}_{i,\mathbf{x}}^{L}$ and $\tilde{\mathbf{v}}_{i,\mathbf{x}}$ encode the same underlying 3D patch i (differing only in their feature extraction pathway), $\beta_{i,\mathbf{x}}$ is equally valid as an importance weight for both.

We also conduct empirical analysis that provides strong evidence supporting this design: $\beta_{i,\mathbf{x}}$ is a stable property of the 3D region for patch i, instead of an artifact of a particular point cloud sample. Specifically, we extract $\beta_{i,\mathbf{x}}$ from 4 normal training samples of each of the 12 categories in the Real3D-AD dataset. To establish patch correspondence across samples (a prerequisite for meaningful comparison, since Farthest Point Sampling [30] produces different patch centers for samples with different orientations), we apply PCA-based canonical alignment [28] to normalize pose, followed by ICP refinement [3] for fine registration. Hungarian matching [17] then establishes one-to-one correspondence between patches across samples based on their aligned 3D center positions. We then measure the agreement of $\beta_{i,\mathbf{x}}$ across all $\binom{4}{2} = 6$ pairwise comparisons within each category using two complementary metrics.

Cosine similarity. Cosine similarity measures the directional agreement between $\beta_{i,\mathbf{x}}$ from two different samples of the same category. It captures whether the relative importance profile across patches is preserved, i.e., whether the same regions are consistently identified as important regardless

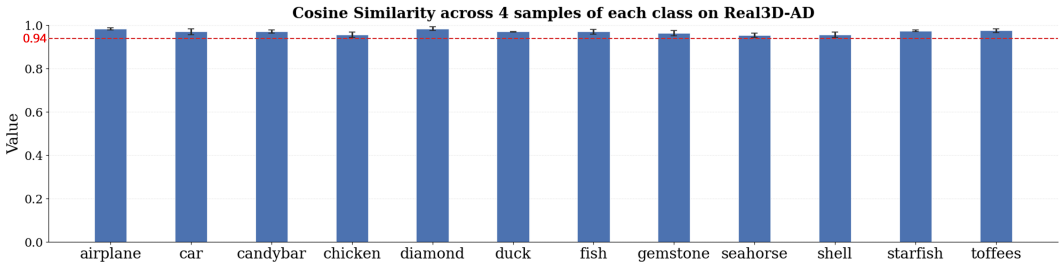


Figure 5: **Cosine similarity of $\beta_{i,\mathbf{x}}$ across normal samples.** Mean pairwise cosine similarity of β vectors computed over all $\binom{4}{2} = 6$ pairs of normal samples per category in the Real3D-AD dataset. Error bars denote standard deviation across pairs. All categories exceed 0.94 (dashed red line), confirming that the importance profile is a stable property of the object geometry.

of the specific point cloud instance. Figure 5 reports the mean pairwise cosine similarity across all 6 pairs for each category. All 12 categories exhibit cosine similarity exceeding 0.94 (dashed red line), with several categories approaching 0.99. The consistently high values and small error bars confirm that the $\beta_{i,\mathbf{x}}$ importance profile is highly stable across different point cloud instances, supporting $\beta_{i,\mathbf{x}}$ as a stable geometric property of the object rather than a sample-specific artifact.

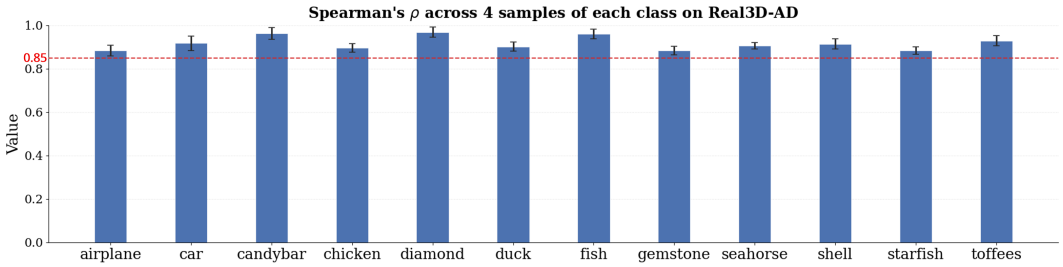


Figure 6: **Spearman's rank correlation of $\beta_{i,\mathbf{x}}$ across normal samples.** Mean pairwise Spearman's ρ of $\beta_{i,\mathbf{x}}$ computed over all $\binom{4}{2} = 6$ pairs of normal samples per category in the Real3D-AD dataset. Error bars denote standard deviation across pairs. All categories exceed 0.85 (dashed red line), confirming that the ordinal ranking of patch importance is well-preserved.

Spearman's rank correlation. While cosine similarity measures directional agreement, Spearman's rank correlation ρ [34] assesses whether the ordinal ranking of patch importance is preserved across samples. This is particularly relevant for our application: in the final combining, the relative ordering of $\beta_{i,\mathbf{x}}$ values determines which regions contribute more to the combined object representation. Figure 6 reports the mean pairwise Spearman's ρ for each category. All categories achieve $\rho > 0.85$ (dashed red line), with several exceeding 0.95. These results confirm that the ordinal ranking of patch importance is well-preserved across point cloud instances: regions that are deemed most (or least) important by $\beta_{i,\mathbf{x}}$ are consistently ranked the same way, regardless of which specific input sample is presented.

Qualitative visualization. To complement the quantitative empirical analysis, Figure 7 provides a qualitative analysis via visualization of $\beta_{i,\mathbf{x}}$ distributions across the 4 normal samples for the Real3D-AD and Anomaly-ShapeNet datasets. Across all four samples within each category, the spatial distribution of high-attention and low-attention regions remains consistent, which corroborates our quantitative empirical analysis findings.

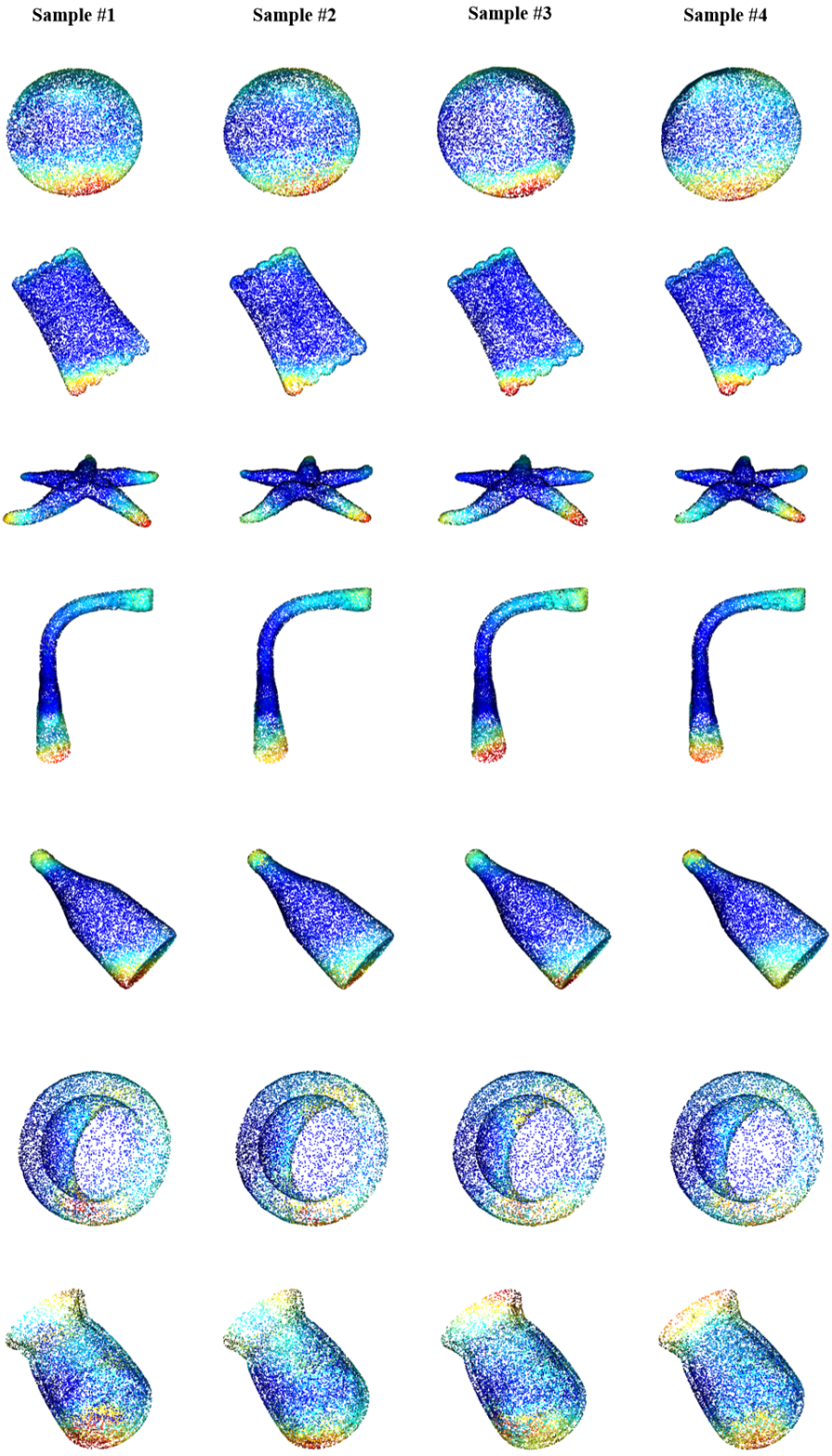


Figure 7: **Visualization of $\beta_{i,\mathbf{x}}$ distributions across normal samples.** The spatial distribution of high-attention regions is consistent across samples within each category, confirming that $\beta_{i,\mathbf{x}}$ captures intrinsic geometric saliency that generalizes across instances.

D Details of GNN in the Combiner

As described in Sec. 3.7, we employ a lightweight Graph Neural Network (GNN) to aggregate the weighted patch embeddings and consolidated point representations while preserving their 3D spatial structure. We provide additional architectural details and sensitivity analysis in this section.

The general design idea for this GNN is *simple* and *lightweight*, where it helps to better capture spatial relationships among patches compared with mean-pooling *without* any complex mechanisms.

Specifically, we construct a k-nearest-neighbor graph over the N=64 patch centers, where each node corresponds to a patch and edges connect spatially proximate patches. We use k_{GNN}=12 by default, so that each node connects to approximately 19% of all patches, providing sufficient spatial coverage of the immediate neighborhood without excessive averaging that could dilute localized anomaly signals. To encode the relative spatial arrangement between connected patches, we compute the relative position for each edge and project it into a d_e-dimensional edge feature via a single linear layer $f_{\text{edge}} : \mathbb{R}^3 \to \mathbb{R}^{d_e}$ with d_e=64. This provides the message-passing operation with explicit awareness of inter-patch direction and distance, complementing the absolute positional information already captured by Point-BERT's positional embeddings [44].

We adopt a single-layer message-passing design for this GNN. For each node i, incoming messages from its k_{GNN} neighbors are formed by concatenating the neighbor's node feature, node i's own feature, and the corresponding edge feature, then transformed by a shared weight matrix W_{GNN}. The aggregated result from the k_{GNN} neighbors by mean pooling is then combined with node i's original feature via a residual connection. We intentionally restrict the GNN to a single layer to avoid over-smoothing [19]. After message passing [11], we apply mean pooling over all N node features to obtain a single d-dimensional vector, which serves as the fused patch-level embedding in Eq. (7).

Table 6: Sensitivity analysis of the GNN neighborhood size k_{GNN}. Best results are **bolded**.

Dataset	Metric	k_{GNN}			
		8	10	12	16
Real3D-AD	O-AUROC	92.6	92.9	**93.1**	92.7
	P-AUROC	93.8	94.3	**94.7**	93.4
Anomaly-ShapeNet	O-AUROC	94.7	94.9	**95.0**	94.8
	P-AUROC	95.1	95.4	**95.6**	94.8

In addition, we conduct a sensitivity analysis of our method to the GNN neighborhood size k_{GNN} by evaluating across $k_{\text{GNN}} \in \{8, 10, 12, 16\}$ on both benchmarks. Note that k_{GNN} and other hyperparameters in our proposed method were selected on a pseudo-validation set constructed via Norm-AS-style synthetic anomalies [43] applied to training samples, and the test set was used only for the final evaluation. We report the sensitivity analysis results of k_{GNN} on the test set solely to characterize robustness, not for hyperparameter selection.

As shown in Table 6, performance peaks at k_{GNN}=12 across both datasets and metrics. The results show that smaller neighborhoods (k_{GNN}=8) limit information flow between spatially related patches, while larger neighborhoods (k_{GNN}=16) lead to over-smoothing that dilutes localized anomaly information. Notably, the performance degradation at k_{GNN}=16 is most pronounced for point-level detection on Real3D-AD. This is because Real3D-AD contains real-world anomalies that tend to be more spatially localized than the synthetic defects in Anomaly-ShapeNet, and point-level evaluation demands precise spatial discrimination.

E Theoretical and Empirical Analysis of Within-Patch Consistency

E.1 Proof of Theorem 1 and Corollary 1

We first introduce the notation and quantities used throughout the analysis (Definition 1), establish dispersion control by $\mathcal{L}_{\text{intra}}$ (Lemma 1), and then prove Theorem 1 and Corollary 1.

Definition 1 (Within-Patch Anomaly Scoring Quantities)**.** *For an input sample* $\mathbf{x}$*, consider a single patch* i *containing* K *points, let* $\hat{\mathbf{v}}_j = \hat{P}\, g^{MLP}(\mathbf{v}_{j,i,\mathbf{x}}) / \|\hat{P}\, g^{MLP}(\mathbf{v}_{j,i,\mathbf{x}})\|_2$ *denote the* ℓ_2*-normalized*

projected feature of point j, and let $\bar{\mathbf{v}}_i = \frac{1}{K}\sum_{k=1}^{K}\hat{\mathbf{v}}_k$ denote the within-patch centroid, with $\|\bar{\mathbf{v}}_i\|_2 > 0$. The local deviation anomaly score of point j is defined as $s_j = 1 - \cos(\hat{\mathbf{v}}_j, \bar{\mathbf{v}}_i) = 1 - (\hat{\mathbf{v}}_j^\top \bar{\mathbf{v}}_i)/\|\bar{\mathbf{v}}_i\|_2$, which is a monotone transformation of s_j^{local} (Eq. (12)) as $s_j = s_j^{local} + 1$ that preserves all rankings and thus the AUROC.

Let one point j^* be anomalous and let the remaining $\mathcal{N} = \{1, \ldots, K\} \setminus \{j^*\}$ be non-anomalous. Let $\bar{s}_{\mathcal{N}} = \frac{1}{|\mathcal{N}|}\sum_{j\in\mathcal{N}} s_j$ denote the mean anomaly score of the non-anomalous points, let $\Delta = s_{j^*} - \bar{s}_{\mathcal{N}}$ denote the anomaly separation, and let $D(j^*) = \Pr(s_{j^*} > \max_{j\in\mathcal{N}} s_j)$ denote the patch-level rank-based detectability, i.e., the probability that the anomalous point achieves the highest score in its patch.

Probability space. All probabilities in this section are taken over the draw of a test patch from the data distribution, including the anomalous point realization, and, for the non-anomalous-score tail condition, over a uniformly sampled non-anomalous point $J \sim \mathrm{Unif}(\mathcal{N})$ conditional on the patch. The conditional boundedness fallback extends to the unconditional bound by taking the expectation over patches because the per-patch conditional parameter is uniformly bounded by 1 under boundedness.

Upper-tail concentration proxy. Let $J \sim \mathrm{Unif}(\mathcal{N})$ be a uniformly sampled non-anomalous point in the patch. We use $\sigma^2_{\mathcal{N},\text{tail}}$ to denote an upper-tail concentration proxy satisfying

$$\Pr(s_J - \bar{s}_{\mathcal{N}} \geq t) \leq \exp\left(-\frac{t^2}{2\sigma^2_{\mathcal{N},\text{tail}}}\right), \quad \text{for all} \quad t > 0, \tag{16}$$

where the probability is over both the patch draw and $J \sim \mathrm{Unif}(\mathcal{N})$.

This condition is placed on the non-anomalous-score (i.e., anomaly score of non-anomalous points) deviations, not on the raw point features.

A loose finite fallback follows from boundedness. Since $s_J - \bar{s}_{\mathcal{N}}$ has support of width at most 2 (conditional on the patch) and $\mathbb{E}_J[s_J \mid \text{patch}] = \bar{s}_{\mathcal{N}}$, Hoeffding's lemma [5] (Lemma 2.2) gives:

$$\mathbb{E}_J\left[\exp(\lambda(s_J - \bar{s}_{\mathcal{N}})) \mid \text{patch}\right] \leq \exp\left(\frac{\lambda^2 \cdot 2^2}{8}\right) = \exp\left(\frac{\lambda^2}{2}\right). \tag{17}$$

Equivalently, by Chernoff–Hoeffding argument [14],

$$\Pr_J(s_J - \bar{s}_{\mathcal{N}} \geq t \mid \text{patch}) \leq \exp\left(-\frac{t^2}{2}\right) = \exp\left(-\frac{t^2}{2\cdot 1}\right). \tag{18}$$

Averaging over test patches gives the same unconditional bound. Hence $\sigma^2_{\mathcal{N},\text{tail}} = 1$ (i.e., $\sigma_{\mathcal{N},\text{tail}} = 1$) is a valid conservative boundedness proxy. This fallback scale is generally too loose to make the numerical lower bound informative, and the useful quantity is instead the empirical fitted tail scale $\widehat{\sigma}_{\mathcal{N},\text{tail}}$ estimated in Appendix E.2. This fitted tail scale should be distinguished from the empirical standard deviation $\widehat{\sigma}_{\mathcal{N},\text{std}}$ reported in Appendix G.4.

Lemma 1 (Dispersion Control by $\mathcal{L}_{\text{intra}}$)**.** *Let $\mathbf{u}_1, \ldots, \mathbf{u}_K \in \mathbb{S}^{\bar{d}-1}$ be generic unit-normalized point representations in a patch of K points, corresponding to $\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K$ in Definition 1, and let $\mathbf{c} = \frac{1}{K}\sum_{j=1}^{K}\mathbf{u}_j$ denote the centroid of point representations, corresponding to $\bar{\mathbf{v}}_i$, with $\|\mathbf{c}\|_2 > 0$. Let $s_j = 1 - \cos(\mathbf{u}_j, \mathbf{c}) = 1 - \mathbf{u}_j^\top \frac{\mathbf{c}}{\|\mathbf{c}\|_2}$ denote the anomaly score for point j and $\bar{s} = \frac{1}{K}\sum_{j=1}^{K} s_j$. Let $\ell_{\text{pair}} = \frac{1}{K(K-1)}\sum_{j\neq k}\left(1 - \mathbf{u}_j^\top \mathbf{u}_k\right)$. Then*

$$\frac{1}{K}\sum_{j=1}^{K}(s_j - \bar{s})^2 \leq \frac{K-1}{K}\ell_{\text{pair}}. \tag{19}$$

Lemma 1 reveals that minimizing $\mathcal{L}_{\text{intra}}$, which is exactly the batch average of the pairwise term ℓ_{pair} after ℓ_2-normalizing the projected point representations inside the cosine, contracts the empirical within-patch score dispersion $\frac{1}{K}\sum_{j=1}^{K}(s_j - \bar{s})^2$ on training patches, and empirically the corresponding test-time dispersion/tail scale decreases with $\mathcal{L}_{\text{intra}}$ (Appendix E.2). Note that patches

in the training phase contain *only* non-anomalous points, so Lemma 1's all-K-point dispersion bound coincides exactly with the non-anomalous dispersion that enters Theorem 1's tail condition. At test time, these two dispersions differ only because the centroid is computed over all K points rather than over $\mathcal{N}$ alone, and Remark 1 shows the resulting per-point score perturbation is $O(1/K)$.

In addition, Lemma 1 is a deterministic variance-dispersion statement. It does not by itself imply the sub-Gaussian-type upper-tail condition in Eq. (16), nor does it analytically identify $\sigma^2_{\mathcal{N},\text{tail}}$ with the empirical variance. Theorem 1 should therefore be read conditionally: if the upper-tail scale of non-anomalous-score deviations is reduced while the anomaly margin is preserved, then patch-level top-rank detectability improves. Since the cosine-distance scores are bounded in $[0, 2]$, Eq. (16) always holds with the conservative fallback $\sigma_{\mathcal{N},\text{tail}} = 1$. Appendix E.2 empirically shows that the fitted tail-decay scale is substantially smaller and decreases with λ_{intra}, consistent with the variance-dispersion mechanism in Lemma 1.

Proof of Lemma 1. Using the pairwise variance identity,

$$\frac{1}{K}\sum_{j=1}^{K}(s_j - \bar{s})^2 = \frac{1}{2K^2}\sum_{j,k}(s_j - s_k)^2. \tag{20}$$

Since $s_j = 1 - \mathbf{u}_j^\top \frac{\mathbf{c}}{\|\mathbf{c}\|_2}$ and $\|\frac{\mathbf{c}}{\|\mathbf{c}\|_2}\|_2 = 1$,

$$|s_j - s_k| = \left|(\mathbf{u}_j - \mathbf{u}_k)^\top \frac{\mathbf{c}}{\|\mathbf{c}\|_2}\right| \le \|\mathbf{u}_j - \mathbf{u}_k\|_2. \tag{21}$$

Therefore,

$$\begin{aligned}\frac{1}{K}\sum_{j=1}^{K}(s_j - \bar{s})^2 &\le \frac{1}{2K^2}\sum_{j,k}\|\mathbf{u}_j - \mathbf{u}_k\|_2^2 \\ &= \frac{1}{2K^2}\sum_{j\neq k} 2(1 - \mathbf{u}_j^\top \mathbf{u}_k) \\ &= \frac{K-1}{K}\ell_{\text{pair}}.\end{aligned} \tag{22}$$

This completes the proof. □

Proof of Theorem 1. Let $\mathcal{E} = \{s_{j^*} \le \max_{j\in\mathcal{N}} s_j\}$ be the ranking failure event. For any deterministic margin $\gamma_\eta > 0$:

$$\Pr(\mathcal{E}) \le \Pr(\Delta < \gamma_\eta) + \Pr(\mathcal{E}, \Delta \ge \gamma_\eta). \tag{23}$$

By the definition of γ_η, $\Pr(\Delta < \gamma_\eta) \le \eta$. On the event $\mathcal{E} \cap \{\Delta \ge \gamma_\eta\}$, there exists $j \in \mathcal{N}$ such that $s_j \ge s_{j^*}$. Hence

$$s_j - \bar{s}_\mathcal{N} \ge s_{j^*} - \bar{s}_\mathcal{N} = \Delta \ge \gamma_\eta. \tag{24}$$

Therefore,

$$\begin{aligned}\Pr(\mathcal{E}, \Delta \ge \gamma_\eta) &\le \Pr(\exists j \in \mathcal{N} : s_j - \bar{s}_\mathcal{N} \ge \gamma_\eta) \\ &\le \mathbb{E}\left[\sum_{j\in\mathcal{N}} \mathbf{1}\left\{s_j - \bar{s}_\mathcal{N} \ge \gamma_\eta\right\}\right] \\ &= (K-1)\mathbb{E}\left[\frac{1}{K-1}\sum_{j\in\mathcal{N}} \mathbf{1}\left\{s_j - \bar{s}_\mathcal{N} \ge \gamma_\eta\right\}\right] \\ &= (K-1)\Pr(s_J - \bar{s}_\mathcal{N} \ge \gamma_\eta) \\ &\le (K-1)\exp\left(-\frac{\gamma_\eta^2}{2\sigma^2_{\mathcal{N},\text{tail}}}\right),\end{aligned} \tag{25}$$

where $J \sim \mathrm{Unif}(\mathcal{N})$ and the last step uses the upper-tail condition in Eq. (16). Combining the two terms gives

$$\Pr(\mathcal{E}) \le \eta + (K-1)\exp\left(-\frac{\gamma_\eta^2}{2\sigma^2_{\mathcal{N},\text{tail}}}\right). \tag{26}$$

Since $D(j^*) = 1 - \Pr(\mathcal{E})$, we obtain

$$D(j^*) \ge 1 - \eta - (K-1)\exp\left(-\frac{\gamma_\eta^2}{2\sigma^2_{\mathcal{N},\text{tail}}}\right), \tag{27}$$

which completes the proof. □

Remark 1 (Test-time centroid perturbation)**.** *Theorem 1 applies to any score variables satisfying the stated margin and upper-tail conditions. Lemma 1 is a deterministic dispersion statement: when applied to any collection of point representations, it controls the score variance computed relative to the centroid of that same collection. During training, all points in each patch are non-anomalous, so Lemma 1 directly controls the all-normal within-patch score dispersion induced by $\mathcal{L}_{intra}$.*

At test time, however, Eq. (12) *computes scores using the all-point centroid, whereas the tail condition in Theorem 1 concerns the score deviations of non-anomalous points. We therefore relate the all-point centroid to the non-anomalous-only centroid. This argument is a deterministic stability calculation; the train-to-test alignment of the resulting tail scale is empirically examined in Appendix E.2 and Appendix G.4.*

Let $\mathcal{A} = \{1,\dots,K\} \setminus \mathcal{N}$ denote the set of anomalous points in the i-th patch and $m = |\mathcal{A}|$ be the number of anomalous points in the i-th patch, and let

$$\bar{\mathbf{v}}_i^{(\mathcal{N})} = \frac{1}{|\mathcal{N}|}\sum_{j\in\mathcal{N}} \hat{\mathbf{v}}_j, \quad \bar{\mathbf{v}}_i^{(\mathcal{A})} = \frac{1}{|\mathcal{A}|}\sum_{j\in\mathcal{A}} \hat{\mathbf{v}}_j. \tag{28}$$

The all-point centroid can be written as

$$\bar{\mathbf{v}}_i = \frac{|\mathcal{N}|}{K}\bar{\mathbf{v}}_i^{(\mathcal{N})} + \frac{|\mathcal{A}|}{K}\bar{\mathbf{v}}_i^{(\mathcal{A})}. \tag{29}$$

Since $\|\bar{\mathbf{v}}_i^{(\mathcal{N})}\|_2 \le 1$, $\|\bar{\mathbf{v}}_i^{(\mathcal{A})}\|_2 \le 1$, and in the single-anomalous point setting of Theorem 1, $m = |\mathcal{A}| = 1$. Therefore, we have

$$\left\|\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_i^{(\mathcal{N})}\right\|_2 = \frac{m}{K}\left\|\bar{\mathbf{v}}_i^{(\mathcal{A})} - \bar{\mathbf{v}}_i^{(\mathcal{N})}\right\|_2 \le \frac{2m}{K} = \frac{2}{K}. \tag{30}$$

Suppose the non-anomalous-only centroid does not collapse:

$$\|\bar{\mathbf{v}}_i^{(\mathcal{N})}\|_2 \ge R \quad \textit{for some} \quad R > \frac{2}{K}. \tag{31}$$

Then every point on the segment between $\bar{\mathbf{v}}_i^{(\mathcal{N})}$ and $\bar{\mathbf{v}}_i$ has norm at least $R - 2/K$. For any unit vector $\mathbf{a}$, the map $\mathbf{c} \mapsto \cos(\mathbf{a},\mathbf{c}) = \mathbf{a}^\top\mathbf{c}/\|\mathbf{c}\|_2$ has derivative norm at most $1/\|\mathbf{c}\|_2$. Hence

$$\begin{aligned}\left|\cos(\mathbf{a},\bar{\mathbf{v}}_i) - \cos(\mathbf{a},\bar{\mathbf{v}}_i^{(\mathcal{N})})\right| &\le \frac{\|\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_i^{(\mathcal{N})}\|_2}{R - 2/K} \\ &\le \frac{2}{K(R-2/K)}.\end{aligned} \tag{32}$$

Thus, each local score is perturbed by at most $2/[K(R-2/K)]$, and the separation $\Delta = s_{j^*} - \bar{s}_{\mathcal{N}}$ is perturbed by at most $4/[K(R-2/K)]$. The qualitative conclusion of Theorem 1 is therefore preserved whenever the margin is large relative to this centroid-direction perturbation. We do not require or claim any numerical value of R. Moreover, for fixed bounded scores, excluding a single point also changes the empirical variance only by $O(1/K)$.

Proof of Corollary 1. Starting from Theorem 1, requiring $D(j^*) \geq 1 - \delta$ is guaranteed if

$$\eta + (K-1)\exp\left(-\frac{\gamma_\eta^2}{2\sigma^2_{\mathcal{N},\text{tail}}}\right) \leq \delta. \tag{33}$$

Since $\eta < \delta$, this is equivalent to

$$(K-1)\exp\left(-\frac{\gamma_\eta^2}{2\sigma^2_{\mathcal{N},\text{tail}}}\right) \leq \delta - \eta. \tag{34}$$

Rearranging it gives

$$\sigma^2_{\mathcal{N},\text{tail}} \leq \frac{\gamma_\eta^2}{2\log\left(\frac{K-1}{\delta-\eta}\right)}. \tag{35}$$

For the reduced-variance claim, suppose the upper-tail proxy is reduced from $\sigma^2_{\mathcal{N},\text{tail}}$ to $(\sigma'_{\mathcal{N},\text{tail}})^2 = \alpha\sigma^2_{\mathcal{N},\text{tail}}$, where $\alpha \in (0,1)$. Setting $\gamma'_\eta = \sqrt{\alpha}\,\gamma_\eta$ preserves the exponent:

$$\frac{(\gamma'_\eta)^2}{2(\sigma'_{\mathcal{N},\text{tail}})^2} = \frac{\alpha\gamma_\eta^2}{2\alpha\sigma^2_{\mathcal{N},\text{tail}}} = \frac{\gamma_\eta^2}{2\sigma^2_{\mathcal{N},\text{tail}}}. \tag{36}$$

Thus, the same detectability guarantee can be maintained with the smaller required margin $\sqrt{\alpha}\,\gamma_\eta$, completing the proof. □

Insights from Theorem 1. Theorem 1 provides two key insights for understanding our point-level anomaly scoring.

1. The detectability depends *exponentially* on $\gamma_\eta^2/\sigma^2_{\mathcal{N},\text{tail}}$, so reducing the tail proxy $\sigma^2_{\mathcal{N},\text{tail}}$ can substantially improve the probability of correctly ranking the anomalous point as the highest-scoring point.
2. The theorem reveals a trade-off: while $\mathcal{L}_{\text{intra}}$ contracts an empirical within-patch score-variance upper bound, excessively strong regularization can collapse useful local geometric variation and reduce the anomaly separation Δ. Appendix G.4 empirically demonstrates this trade-off using the empirical standard deviation $\widehat{\sigma}_{\mathcal{N},\text{std}}$: this empirical standard deviation decreases monotonically as λ_{intra} increases, while Δ remains stable for small λ_{intra} but collapses at larger values ($\lambda_{\text{intra}} \geq 0.2$), with both the empirical signal-to-noise ratio (SNR) proxy and P-AUROC peaking at $\lambda_{\text{intra}} = 0.1$.

Insights from Corollary 1. Corollary 1 reveals that a tighter upper-tail proxy (i.e., smaller $\sigma^2_{\mathcal{N},\text{tail}}$) allows detection of subtler anomalies: reducing $\sigma^2_{\mathcal{N},\text{tail}}$ by a factor α reduces the required margin from γ_η to $\sqrt{\alpha}\gamma_\eta$ for the same exponential term. This explains how $\mathcal{L}_{\text{intra}}$ improves sensitivity to fine-grained point-level defects.

Moreover, the within-patch consistency principle is applicable because FACETS specifically retains point-level features from the tokenizer's (i.e., Mini-PointNet) pre-max-pooling representations.

E.2 Empirical Illustration of Theorem 1

Beyond the theoretical results in Appendix E.1, we further conduct empirical analysis for Theorem 1 on Real3D-AD and Anomaly-ShapeNet datasets across $\lambda_{\text{intra}} \in \{0.01, 0.05, 0.1, 0.2\}$ for two aspects[3]: (i) the empirical upper-tail behavior of non-anomalous-scores, and (ii) the exponential trend characterized in Theorem 1.

Empirical concentration behavior of non-anomalous-scores. For each non-anomalous point j in every patch of the test set, we compute the score deviation $s_j - \bar{s}_{\mathcal{N}}$, pool these deviations over all test patches (without per-patch standardization), and measure their empirical complementary

[3]For readability, Figure 8 visualizes the main non-overregularized range $\lambda_{\text{intra}} \in \{0.01, 0.05, 0.1, 0.2\}$; the full sensitivity sweep including $\lambda_{\text{intra}} = 0$ and $\lambda_{\text{intra}} = 0.5$ is reported in Appendix G.4.

CDF $\widehat{\Pr}(s_j - \bar{s}_{\mathcal{N}} > t)$. Figure 8 plots $\log \widehat{\Pr}(s_j - \bar{s}_{\mathcal{N}} > t)$ against t^2 for both datasets. Under the upper-tail concentration model in Eq. (16), this relationship is expected to be approximately linear with slope $-1/(2\sigma^2_{\mathcal{N},\text{tail}})$. Across all tested λ_{intra} values and both datasets, the empirical curves approximately follow this linear trend, which is consistent with the upper-tail model used in Theorem 1. We obtain $\hat{\sigma}_{\mathcal{N},\text{tail}}$ from an ordinary least-squares fit (with intercept) of this log-CCDF on t^2 over all $t > 0$ at which the empirical CCDF is positive, setting its slope equal to $-1/(2\hat{\sigma}^2_{\mathcal{N},\text{tail}})$. For readability, Figure 8 shows the empirical CCDF only down to about 10^{-3}, with least-squares lines fitted to the displayed points; the tail beyond this range decays more slowly, so the full-range $\hat{\sigma}_{\mathcal{N},\text{tail}}$ in Table 7 is somewhat larger than the scale implied by the displayed slopes. It is nevertheless much smaller than the loose boundedness fallback scale $\sigma_{\mathcal{N},\text{tail}} = 1$ and decreases with λ_{intra}, consistent with the dispersion control in Lemma 1.

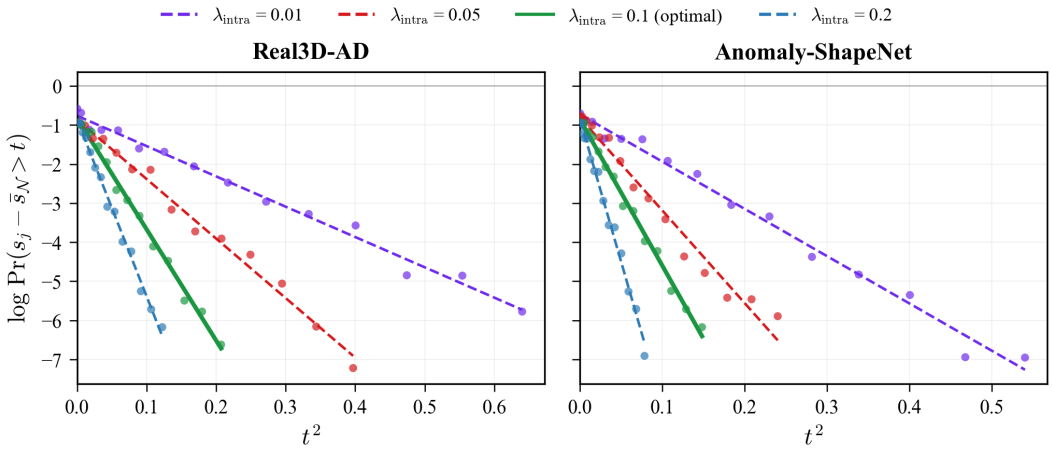


Figure 8: **Empirical concentration behavior of non-anomalous-scores.** Empirical $\log \Pr(s_j - \bar{s}_{\mathcal{N}} > t)$ versus t^2 for non-anomalous points across different λ_{intra} values on Real3D-AD (left) and Anomaly-ShapeNet (right). The approximately linear relationships are consistent with the upper-tail concentration model used in Theorem 1. Markers show the empirical CCDF down to about 10^{-3} for readability; lines are least-squares fits (with intercept) to the displayed markers, whereas $\widehat{\sigma}_{\mathcal{N},\text{tail}}$ in Table 7 is fitted over the full range.

Why use $\hat{\sigma}_{\mathcal{N},\text{std}}$ in empirical diagnostics? Theorem 1 is stated in terms of the upper-tail scale $\sigma_{\mathcal{N},\text{tail}}$, which controls the concentration of non-anomalous-score deviations. In the empirical analysis, we additionally report the standard-deviation scale $\hat{\sigma}_{\mathcal{N},\text{std}}$, computed directly from non-anomalous-scores within each patch. This quantity is not a formal finite-sample estimator of $\sigma_{\mathcal{N},\text{tail}}$; rather, it is a stable and directly interpretable measure of within-patch non-anomalous-score dispersion. We use it in two diagnostic ways: first, as a plug-in dispersion scale for the qualitative exponential trend in Figure 9, and second, as the denominator of the empirical SNR proxy $\Delta/\hat{\sigma}_{\mathcal{N},\text{std}}$ in Appendix G.4. Empirically, the fitted tail scale $\hat{\sigma}_{\mathcal{N},\text{tail}}$ (the full-range CCDF fit above) decreases monotonically together with $\hat{\sigma}_{\mathcal{N},\text{std}}$ across the λ_{intra} sweep, as summarized in Table 7. Thus, $\hat{\sigma}_{\mathcal{N},\text{std}}$ preserves the qualitative margin-to-dispersion trend suggested by Theorem 1.

Empirical detectability and the exponential trend. To examine the exponential dependence suggested by Theorem 1, we measure the empirical detectability $D(j^*) = \Pr(s_{j^*} > \max_{j \in \mathcal{N}} s_j)$ together with the empirical separation Δ and the empirical non-anomalous-score standard deviation $\widehat{\sigma}_{\mathcal{N},\text{std}}$ for each λ_{intra} setting. For patches containing multiple anomalous points, let $\mathcal{A}$ denote the set of anomalous points and let $\mathcal{N}$ denote the set of non-anomalous points. We define $j^* \in \arg\max_{a \in \mathcal{A}} s_a$ and compute all non-anomalous statistics, including $\bar{s}_{\mathcal{N}}$, $\widehat{\sigma}_{\mathcal{N},\text{std}}$, and the empirical upper-tail curve, using points in $\mathcal{N}$ only. Thus, other anomalous points are never included in the non-anomalous tail distribution. The resulting detectability quantity measures whether at least one anomalous point outranks all non-anomalous points in the patch: $\Pr(s_{j^*} > \max_{j \in \mathcal{N}} s_j) = \Pr(\max_{a \in \mathcal{A}} s_a > \max_{j \in \mathcal{N}} s_j)$. Figure 9(i) plots empirical $D(j^*)$ against $\Delta^2/(2\widehat{\sigma}^2_{\mathcal{N},\text{std}})$, overlaid with the exponential curve $1-(K-1)\exp(-\Delta^2/(2\widehat{\sigma}^2_{\mathcal{N},\text{std}}))$. Because Theorem 1 is stated in terms of

Table 7: Monotone alignment between the empirical standard deviation $\widehat{\sigma}_{\mathcal{N},\text{std}}$ and the fitted upper-tail scale $\widehat{\sigma}_{\mathcal{N},\text{tail}}$ across λ_{intra} sweep on Real3D-AD and Anomaly-ShapeNet datasets. Both $\widehat{\sigma}_{\mathcal{N},\text{std}}$ and $\widehat{\sigma}_{\mathcal{N},\text{tail}}$ decrease monotonically with λ_{intra} on both datasets. $\widehat{\sigma}_{\mathcal{N},\text{tail}}$ is fitted over the full range of the pooled empirical CCDF, beyond the range displayed in Figure 8; the curves for $\lambda_{\text{intra}} \in \{0, 0.5\}$ are omitted from Figure 8 only for readability.

Dataset	**Quantity**	λ_{intra}					
		0	0.01	0.05	0.1	0.2	0.5
Real3D-AD	$\widehat{\sigma}_{\mathcal{N},\text{std}}$	0.34	0.25	0.18	0.13	0.10	0.07
	$\widehat{\sigma}_{\mathcal{N},\text{tail}}$	0.43	0.32	0.23	0.16	0.12	0.08
Anomaly-ShapeNet	$\widehat{\sigma}_{\mathcal{N},\text{std}}$	0.29	0.21	0.14	0.11	0.08	0.06
	$\widehat{\sigma}_{\mathcal{N},\text{tail}}$	0.37	0.27	0.18	0.14	0.10	0.07

a deterministic high-probability margin γ_η and a valid upper-tail scale $\sigma_{\mathcal{N},\text{tail}}$, whereas this plot uses empirical separation statistics and $\widehat{\sigma}_{\mathcal{N},\text{std}}$, this curve should be interpreted as a plug-in visualization using empirical separation statistics, rather than as a formal finite-sample confidence bound. On both datasets, the empirical detectability follows the same characteristic trend: detectability increases as the margin-to-dispersion ratio grows, peaks near $\lambda_{\text{intra}} = 0.1$, and declines at larger values as the separation Δ collapses, consistent with the trade-off analyzed in Figure 13 of Appendix G.4. Figure 9(ii) further plots $\log(1 - D(j^*))$ against $\Delta^2/(2\widehat{\sigma}^2_{\mathcal{N},\text{std}})$, which linearizes the plug-in exponential relation. The approximately linear downward trends are consistent with the exponential dependence predicted by Theorem 1.

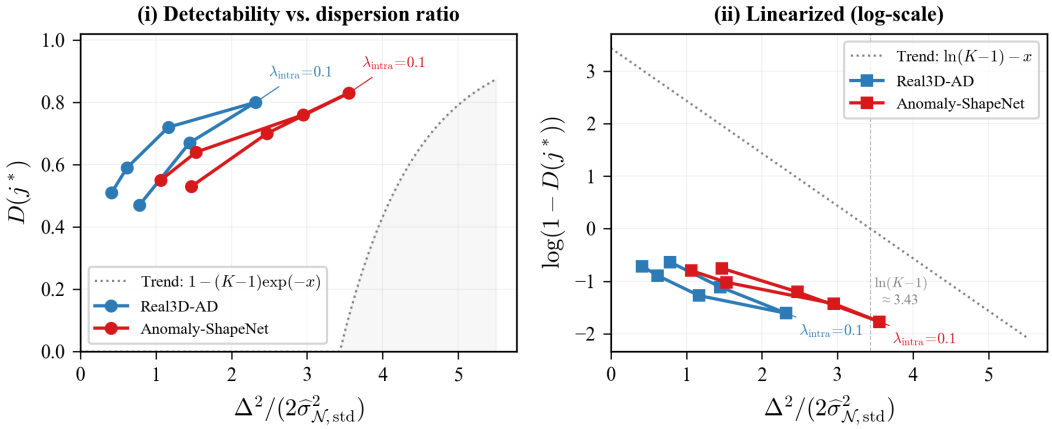


Figure 9: **Empirical detectability and the exponential trend.** (i) Empirical detectability $D(j^*)$ versus $\Delta^2/(2\widehat{\sigma}^2_{\mathcal{N},\text{std}})$ on Real3D-AD and Anomaly-ShapeNet, with the exponential trend shown as a dotted gray curve. Each point along a curve corresponds to a different λ_{intra} value; the optimal $\lambda_{\text{intra}} = 0.1$ is annotated. (ii) Linearized view: $\log(1 - D(j^*))$ versus $\Delta^2/(2\widehat{\sigma}^2_{\mathcal{N},\text{std}})$; the approximately linear downward trends are consistent with the exponential dependence in Theorem 1. The vertical dashed line marks $\ln(K - 1) \approx 3.43$, the plug-in threshold above which the exponential curve becomes non-vacuous with $\eta = 0$.

F Details of Implementation

In this section, we provide detailed information on our implementation. We implement our method using PyTorch 2.8.0 in Python 3.11. By default, all experiments are executed on a four-GPU NVIDIA RTX 5090 workstation. To assess the efficiency and scalability of our method in a more accessible hardware environment, we additionally evaluate it on a GPU workstation equipped with a single NVIDIA RTX 3090 GPU.

Our framework introduces only two task-specific hyperparameters: (i) the GNN neighborhood size k_{GNN}=12, which connects each patch to $\sim$19% of all N=64 patches to balance spatial connectivity with locality preservation (sensitivity analysis available in Appendix D); and (ii) the intra-patch consistency loss weight λ_{intra}=0.1, which balances within-patch coherence regularization against over-constraining the point embeddings (sensitivity analysis available in Appendix G.4).

As for the 3D VLM backbone, we choose the standard setup of ULIP-2 with Point-BERT, as described in Sec. 5. Specifically, there are two publicly available checkpoints for ULIP-2 with Point-BERT: one pre-trained on the Objaverse dataset only and the other one pre-trained on both the Objaverse and ShapeNet datasets. We use the *first* one (i.e., the one pre-trained *only* on the Objaverse dataset) throughout all experiments in our study to ensure a fair performance comparison with competing approaches, as it has no overlap between its pre-training data and the Anomaly-ShapeNet dataset. Our framework keeps both the 3D encoder and the text encoder frozen, as well as the original ULIP-2 projection matrix $P \in \mathbb{R}^{\bar{d}\times 2d}$.

Patches are constructed with the standard Point-BERT/ULIP-2 tokenization protocol: N=64 patch centers are selected from each input point cloud by farthest point sampling, and the K=32 nearest neighbors of each center form its patch $\mathcal{P}_{i,\mathbf{x}}$. Patches may overlap, and the coordinates within each patch are normalized relative to its center before tokenization. This is the tokenization used in the pre-training of the frozen backbone, which keeps the extracted patch- and point-level features compatible with ULIP-2's pre-trained representation space. As for the shared learnable projection layer $\hat{P} \in \mathbb{R}^{\bar{d}\times d}$, it is initialized from $P^{\text{pool}} = P_{[:,d:2d]}$, the submatrix of the frozen ULIP-2 projection that corresponds to the max-pooled patch features. This initialization leverages the pre-trained alignment between patch-level features and the joint embedding space, and fine-tuning allows gradual adaptation from the max-pooled feature distribution to the individual patch/point feature distribution (Eq. (3) writes $\hat{P}$ for every i for brevity; for the `[CLS]` output embedding $\mathbf{z}^L_{0,\mathbf{x}}$, we use the submatrix $P^{\text{CLS}} = P_{[:,0:d]}$, which is kept frozen throughout the entire training procedure). For the MLP $g^{\text{MLP}} : \mathbb{R}^{d_{\text{mini}}} \to \mathbb{R}^d$ (where d_{mini} is the dimension of the native point-level features produced by the tokenizer before max-pooling, i.e., $\mathbf{v}_{j,i,\mathbf{x}} \in \mathbb{R}^{d_{\text{mini}}}$), it is a two-layer MLP with ReLU activation that maps pre-max-pooling point features from tokenizer (Mini-PointNet) into the patch embedding space. Weights are initialized with Kaiming uniform initialization [13]. For the GNN in the combiner, we use a single-layer graph convolution $W_{\text{GNN}} \in \mathbb{R}^{d\times(2d+d_e)}$ following the EdgeConv-style [39] formulation with residual connection (more details on its design rationale are available in Appendix D). Weights are initialized with Xavier uniform initialization [12]. The edge encoding in GNN uses a single linear layer $f_{\text{edge}} : \mathbb{R}^3 \to \mathbb{R}^{d_e}$ initialized with Xavier uniform initialization. The temperature τ is learnable and initialized to 0.07.

We train a single, unified model per dataset in a multi-category fashion (i.e., jointly across all categories), rather than maintaining independent per-category models, as is common in most prior 3D anomaly detection methods. The optimizer is AdamW [25] with a base learning rate of 5×10^{-4} and weight decay of 0.01. We apply a differential learning rate strategy: the learnable projection layer $\hat{P}$ uses a $10\times$ lower learning rate (5×10^{-5}) to preserve its pre-trained initialization while allowing gradual adaptation. The learning rate schedule consists of a linear warmup over the first 5 epochs, followed by cosine annealing [24]. We set the mini-batch size to 128 and train our framework for 200 epochs with gradient clipping at a maximum norm of 1.0. Standard 3D point cloud augmentations are applied during training, including random rotation and random scaling.

Each input point cloud is randomly resampled to N_p=8,192 points, in training and at test time. At test time, the point-level score of Sec. 3.9 is computed for every point of every patch. The combined score s_j^{point} is standardized within each patch (zero mean and unit variance over its K points), and a point that belongs to several patches receives the unweighted arithmetic mean of its standardized per-patch scores. Every point of the full-resolution ground truth then receives the inverse-distance-weighted average of the scores of its three nearest scored points.

G Supplementary Results

G.1 Extended Main Results

In this section, we provide the complete per-category quantitative evaluation results and additional qualitative visualizations for the main results presented in Sec. 5.2.

Per-Category Quantitative Results. Tables 8 and 9 present the per-category breakdown of object-level AUROC (O-AUROC) and point-level AUROC (P-AUROC) on Real3D-AD and Anomaly-ShapeNet datasets, respectively.

Real3D-AD. As shown in Table 8, our method achieves the best (or tied-best) O-AUROC in 6 out of 12 categories and the best P-AUROC in 10 out of 12 categories. It is second-best in the remaining categories, except for *Diamond*, *Gemstone*, and *Shell* (O-AUROC), where it ranks third. Notably, in categories where prior methods exhibit significant performance variance, such as *Duck* and *Car*, our method consistently achieves top-tier performance. This stability across diverse object geometries highlights the effectiveness of our proposed mechanisms in FACETS. It is also worth noting that certain methods achieve strong results on individual categories but exhibit high variance across the full benchmark; for instance, RPRIL [45] attains 100.0% O-AUROC on *Candybar* and *Diamond* but drops to 53.2% on *Seahorse* and 62.6% on *Toffees*. In contrast, our method maintains more balanced performance, with its lowest per-category O-AUROC (79.3% on *Gemstone*) still exceeding the dataset-wide average of most competing methods.

Anomaly-ShapeNet. Table 9 presents the per-category results on the Anomaly-ShapeNet dataset. Our method achieves the best (or tied-best) O-AUROC in 22 out of 40 categories and the best P-AUROC in 21 out of 40 categories, demonstrating strong generalization across a much larger category space. It is second-best in the remaining categories, except for *cap0*, *cup1*, *shelf0*, *tap0*, and *tap1* (O-AUROC) and *bottle1* (P-AUROC), where it ranks third. Our method also leads in categories where prior methods struggle. For example, on *helmet0*, where 14 of the 16 other methods that report O-AUROC fall below 80%, our method reaches 83.2%/95.4% compared to the next best 81.7%/92.5%; on *cap5*, it reaches 93.1%/98.4% compared to 90.2%/97.0%.

Additional Qualitative Results. Figure 10 presents additional anomaly map visualizations on both Real3D-AD (top two rows) and Anomaly-ShapeNet (bottom two rows). On the Real3D-AD dataset, our method accurately localizes anomalous regions across objects with diverse geometries. On the Anomaly-ShapeNet dataset, similar localization quality is observed for synthetic anomalies on those objects. Notably, the predicted anomaly maps exhibit well-concentrated high-score regions that closely align with the ground truth defect areas, with minimal false positive activations on normal surfaces.

Comparison with Native 3D-VLM Methods. Table 10 compares FACETS with PLANE [38] and BTP [18] (Appendix A.3), using the results reported in their papers; both are also included in Tables 1, 8, and 9. PLANE follows the same unified multi-class, unsupervised protocol on the same ULIP-2 backbone, and FACETS outperforms it by 15.4/21.1 points (O-/P-AUROC) on Real3D-AD and 11.4/13.5 points on Anomaly-ShapeNet. BTP follows a zero-shot protocol (trained on one labeled auxiliary category, with no target-category data), so its results are listed for completeness and are not directly comparable.

G.2 Visualization of Our Weights $\beta_{i,\mathbf{x}}$, $\mu^{(\mathbf{z})}_{i,\mathbf{x}}$, and $\mu^{(\mathbf{v})}_{i,\mathbf{x}}$

To gain deeper insight into the roles of the holistic–local attention weight $\beta_{i,\mathbf{x}}$, the patch linguistic grounding weight $\mu^{(\mathbf{z})}_{i,\mathbf{x}}$, and the consolidated-point linguistic grounding weight $\mu^{(\mathbf{v})}_{i,\mathbf{x}}$ in our framework, we visualize them on normal point cloud samples from both Real3D-AD [23] and Anomaly-ShapeNet [20] datasets. Figure 11 presents the visualization with warm colors (red/yellow) indicating high values and cool colors (blue) indicating low values.

$\beta_{i,\mathbf{x}}$ captures geometrically salient regions. For $\beta_{i,\mathbf{x}}$, the top row of Figure 11 shows that $\beta_{i,\mathbf{x}}$ consistently assigns high weight to patches located at geometrically distinctive regions of each object, such as edges, thin protrusions, high-curvature surfaces, and structural boundaries. For example, the tail and fin extremities of *Fish* and *Seahorse*, the brim edge of *cap*, the handle of *cup*, and the narrow neck of *vase* all receive notably higher $\beta_{i,\mathbf{x}}$ values, while flat, featureless surfaces remain

Table 8: Comparison of object-level AUROC (O-AUROC, ↑) and point-level AUROC (P-AUROC, ↑) of competing methods on Real3D-AD dataset. Each cell shows O-AUROC / P-AUROC. **Bold** indicates best, underline indicates second best per metric. "–" indicates results not available in the original paper or publicly available implementation under a directly comparable setting. Methods marked with † use a zero-shot protocol (no training data from the target categories).

Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish
BTF (RAW) [15]	73.0 / 56.4	64.7 / 64.7	53.9 / 73.5	78.9 / 60.9	70.7 / 56.3	69.1 / 60.1	60.2 / 51.4
BTF (FPFH)	52.0 / 73.8	56.0 / 70.8	63.0 / 86.4	43.2 / 73.5	54.5 / 88.2	78.4 / 87.5	54.9 / 70.9
M3DM [40]	43.4 / 54.7	54.1 / 60.2	55.2 / 67.9	68.3 / 67.8	60.2 / 60.8	43.3 / 66.7	54.0 / 60.6
PatchCore (FPFH) [33]	88.2 / 56.2	59.0 / 75.4	54.1 / 78.0	83.7 / 42.9	57.4 / 82.8	54.6 / 26.4	67.5 / 82.9
PatchCore (PMAE)	72.6 / 56.9	49.8 / 60.9	66.3 / 62.7	82.7 / 72.9	78.3 / 71.8	48.9 / 52.8	63.0 / 71.7
CPMF [6]	70.1 / 61.8	55.1 / 83.6	55.2 / 73.4	50.4 / 55.9	52.3 / 75.3	58.2 / 71.9	55.8 / **98.8**
Reg3D-AD [23]	71.6 / 63.1	69.7 / 71.8	68.5 / 72.4	85.2 / 67.6	90.0 / 83.5	58.4 / 50.3	91.5 / 82.6
Group3AD [51]	74.4 / 63.6	72.8 / 74.5	84.7 / 73.8	78.6 / 75.9	93.2 / 86.2	67.9 / 63.1	97.6 / 83.6
IMRNet [20]	76.2 / –	71.1 / –	75.5 / –	78.0 / –	90.5 / –	51.7 / –	88.0 / –
R3D-AD [50]	77.2 / –	69.3 / –	71.3 / –	71.4 / –	68.5 / –	90.9 / –	69.2 / –
PointAD† [49]	60.9 / 67.2	73.9 / 72.3	74.1 / 71.3	52.0 / 67.7	99.2 / 87.7	60.1 / 51.0	74.3 / 80.1
ISMP [22]	85.8 / 75.3	73.1 / 83.6	85.2 / 90.7	71.4 / 79.8	94.8 / 92.6	71.2 / 87.6	94.5 / 88.6
PO3AD [43]	80.4 / –	65.4 / –	78.5 / –	68.6 / –	80.1 / –	82.0 / –	85.9 / –
RPRIL [45]	81.8 / 92.3	75.8 / 94.4	**100.0** / 96.9	94.4 / 91.0	**100.0** / 97.9	75.0 / 93.7	67.2 / 84.6
PASDF [47]	62.8 / 77.7	**95.9** / 80.2	78.8 / 54.6	73.9 / 76.8	89.4 / 69.9	65.8 / 78.2	98.9 / 83.7
CASL [46]	80.8 / 84.2	79.9 / 90.5	84.8 / 93.2	65.7 / 71.3	97.6 / 98.8	83.6 / 89.5	93.5 / 93.5
MC3D-AD [8]	85.0 / 62.8	74.9 / 81.9	83.0 / 91.0	71.5 / 64.0	95.5 / 94.2	83.1 / 82.2	86.5 / 93.2
PLANE [38]	63.0 / 69.2	77.4 / 67.8	83.3 / 83.0	69.3 / 58.0	99.8 / 90.7	83.9 / 68.6	93.9 / 83.0
BTP† [18]	42.9 / 75.9	55.9 / 91.6	56.3 / 77.7	48.5 / 85.0	94.4 / 97.9	66.7 / 90.9	59.6 / 90.3
Ours	**90.7**$_{\pm 0.1}$ / **93.7**$_{\pm 0.2}$	94.3$_{\pm 0.4}$ / **95.6**$_{\pm 0.3}$	98.1$_{\pm 0.2}$ / **98.2**$_{\pm 0.2}$	**95.8**$_{\pm 0.2}$ / **93.9**$_{\pm 0.4}$	99.6$_{\pm 0.3}$ / **99.7**$_{\pm 0.1}$	**92.5**$_{\pm 0.2}$ / **95.8**$_{\pm 0.1}$	**99.7**$_{\pm 0.1}$ / 95.2$_{\pm 0.5}$

Method	Gemstone	Seahorse	Shell	Starfish	Toffees	*Average*
BTF (RAW) [15]	68.6 / 59.7	59.6 / 52.0	39.6 / 48.9	53.0 / 39.2	70.3 / 62.3	63.5 / 57.1
BTF (FPFH)	64.8 / 89.1	77.9 / 51.2	75.4 / 57.1	57.5 / 50.1	46.2 / 81.5	60.3 / 73.3
M3DM [40]	64.4 / 67.4	49.5 / 56.0	69.4 / 73.8	55.1 / 53.2	45.0 / 68.2	55.2 / 63.1
PatchCore (FPFH) [33]	37.0 / 91.0	50.5 / 73.9	58.9 / 73.9	44.1 / 60.6	56.5 / 74.7	59.3 / 68.2
PatchCore (PMAE)	37.4 / 44.4	53.9 / 63.3	50.1 / 70.9	51.9 / 58.0	58.5 / 58.0	59.5 / 62.0
CPMF [6]	58.9 / 44.9	72.9 / 96.2	65.3 / 72.5	70.0 / 80.0	39.0 / 95.9	58.6 / 75.9
Reg3D-AD [23]	41.7 / 54.5	76.2 / 81.7	58.3 / 81.1	50.6 / 61.7	82.7 / 75.9	70.4 / 70.5
Group3AD [51]	53.9 / 56.4	84.1 / 82.7	58.5 / 79.8	56.2 / 62.5	79.6 / 80.3	75.1 / 73.5
IMRNet [20]	67.4 / –	60.4 / –	66.5 / –	67.4 / –	77.4 / –	72.5 / –
R3D-AD [50]	66.5 / –	72.0 / –	84.0 / –	70.1 / –	70.3 / –	73.4 / –
PointAD† [49]	87.3 / 80.2	76.9 / 74.8	**89.5** / 77.8	80.9 / 81.4	69.0 / 70.0	74.8 / 73.5
ISMP [22]	46.8 / 85.7	72.9 / 81.3	62.3 / 83.9	66.0 / 64.1	84.2 / 89.5	75.7 / 83.6
PO3AD [43]	69.3 / –	75.6 / –	80.0 / –	75.8 / –	77.1 / –	76.6 / –
RPRIL [45]	73.5 / 90.7	53.2 / 64.5	69.2 / **90.6**	84.1 / 84.0	62.6 / 73.7	78.1 / 87.9
PASDF [47]	63.4 / 65.4	**100.0** / 88.7	85.0 / 64.8	61.7 / 70.3	86.6 / 83.8	80.2 / 74.5
CASL [46]	76.9 / 91.6	64.3 / 81.4	79.1 / 87.3	**89.3** / 83.9	92.4 / 93.7	82.3 / 88.2
MC3D-AD [8]	56.0 / 45.8	71.6 / 65.9	80.3 / 77.8	76.6 / 69.0	73.8 / 93.4	78.2 / 76.8
PLANE [38]	**89.1** / 83.5	54.0 / 64.2	86.6 / 74.7	60.1 / 59.2	71.6 / 81.9	77.7 / 73.6
BTP† [18]	71.7 / 87.2	53.5 / 68.8	73.6 / 81.3	38.3 / 74.5	74.9 / 93.2	61.4 / 84.5
Ours	79.3$_{\pm 0.3}$ / **93.1**$_{\pm 0.1}$	**100.0**$_{\pm 0.0}$ / **97.8**$_{\pm 0.2}$	86.5$_{\pm 0.4}$ / 89.0$_{\pm 0.4}$	86.6$_{\pm 0.2}$ / **86.3**$_{\pm 0.3}$	**94.1**$_{\pm 0.3}$ / **97.7**$_{\pm 0.1}$	**93.1**$_{\pm 0.2}$ / **94.7**$_{\pm 0.2}$

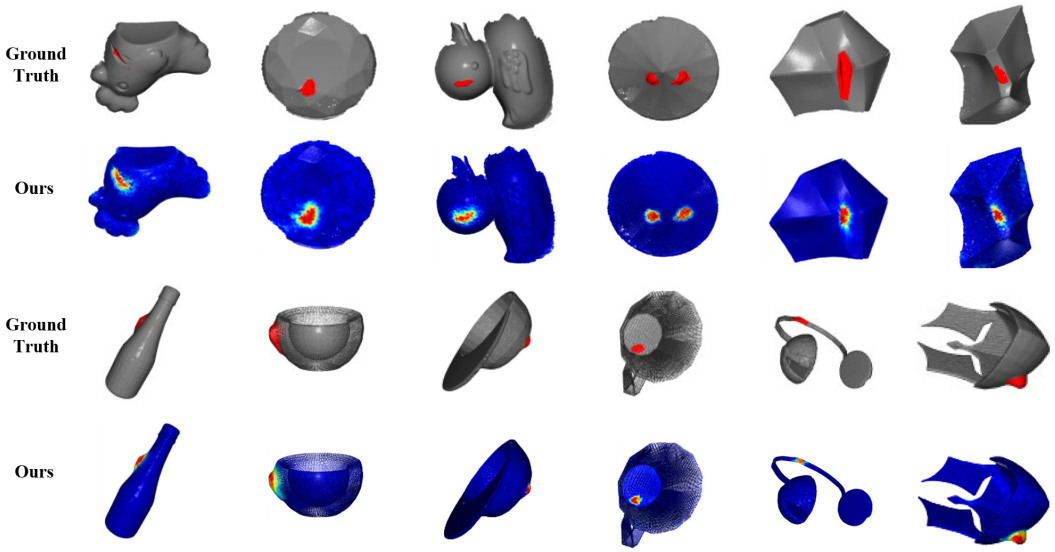


Figure 10: Additional anomaly map visualizations on the Real3D-AD and Anomaly-ShapeNet datasets.

Table 9: Comparison of O-AUROC and P-AUROC results (↑) of competing methods on Anomaly-ShapeNet. Each cell shows O-AUROC / P-AUROC. **Bold** indicates best, underline indicates second best per metric. "–" indicates results not available in the original paper or publicly available implementation under a directly comparable setting. Methods marked with † use a zero-shot protocol (no training data from the target categories).

Method	ashtray0	bag0	bottle0	bottle1	bottle3	bowl0	bowl1	bowl2	bowl3	bowl4	bowl5
BTF (RAW) [15]	57.8 / 51.2	41.0 / 43.0	59.7 / 55.1	51.0 / 49.1	56.8 / 72.0	56.4 / 52.4	26.4 / 46.4	52.5 / 42.6	38.5 / 68.5	66.4 / 56.3	41.7 / 51.7
BTF (FPFH)	42.0 / 62.4	54.6 / 74.6	34.4 / 64.1	54.6 / 54.9	32.2 / 62.2	50.9 / 71.0	66.8 / 76.8	51.0 / 51.8	49.0 / 59.0	60.9 / 67.9	69.9 / 69.9
M3DM [40]	57.7 / 57.7	53.7 / 63.7	57.4 / 66.3	63.7 / 63.7	54.1 / 53.2	63.4 / 65.8	66.3 / 66.3	68.4 / 69.4	61.7 / 65.7	46.4 / 62.4	40.9 / 48.9
PatchCore (FPFH) [33]	58.7 / 59.7	57.1 / 57.4	60.4 / 65.4	66.7 / 68.7	57.2 / 51.2	50.4 / 52.4	63.9 / 53.1	61.5 / 62.5	53.7 / 32.7	49.4 / 72.0	55.8 / 35.8
PatchCore (PMAE)	59.1 / 49.5	60.1 / 67.4	51.3 / 55.3	60.1 / 60.6	65.0 / 65.3	52.3 / 52.7	62.9 / 52.4	45.8 / 51.5	57.9 / 58.1	50.1 / 50.1	59.3 / 56.2
CPMF [6]	35.3 / 61.5	64.3 / 65.5	52.0 / 52.1	48.2 / 57.1	40.5 / 43.5	78.3 / 74.5	63.9 / 48.8	62.5 / 63.5	65.8 / 64.1	68.3 / 68.3	68.5 / 68.4
Reg3D-AD [23]	59.7 / 69.8	70.6 / 71.5	48.6 / 88.6	69.5 / 69.6	52.5 / 52.5	67.1 / 77.5	52.5 / 61.5	49.0 / 59.3	34.8 / 65.4	66.3 / 80.0	59.3 / 69.1
Group3AD [51]	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –
IMRNet [20]	67.1 / 67.1	66.0 / 66.8	55.2 / 55.6	70.0 / 70.2	64.0 / 64.1	68.1 / 78.1	70.2 / 70.5	68.5 / 68.4	59.9 / 59.9	67.6 / 57.6	71.0 / 71.5
R3D-AD [50]	83.3 / –	72.0 / –	73.3 / –	73.7 / –	78.1 / –	81.9 / –	77.8 / –	74.1 / –	76.7 / –	74.4 / –	65.6 / –
PointAD† [49]	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –
ISMP [22]	– / 60.3	– / 74.7	– / 77.0	– / 56.8	– / 77.5	– / 85.1	– / 54.6	– / 73.6	– / 77.3	– / 74.0	– / 53.4
PO3AD [43]	**100.0** / 96.2	83.3 / 94.9	90.0 / 91.2	93.3 / 84.4	92.6 / 88.0	92.2 / 97.8	82.9 / 91.4	83.3 / 91.8	88.1 / 93.5	98.1 / 96.9	84.9 / 94.1
RPRIL [45]	90.0 / 78.5	**100.0** / 99.1	**100.0** / 99.5	**100.0** / 84.9	**100.0** / 81.7	**100.0** / 98.3	80.7 / 82.8	65.6 / 82.2	58.5 / 76.1	85.2 / 78.8	81.8 / 82.4
PASDF [47]	**100.0** / 91.9	99.5 / 95.8	**100.0** / 95.1	**100.0** / **92.6**	**100.0** / 94.8	**100.0** / 96.3	94.8 / 90.0	**100.0** / 81.6	**100.0** / 93.9	93.3 / 86.5	91.2 / 90.9
CASL [46]	94.3 / 88.7	94.8 / 96.8	95.7 / 85.3	95.4 / 86.2	**100.0** / 91.4	**100.0** / 95.8	93.3 / **97.5**	99.3 / **98.2**	99.6 / **99.8**	94.4 / 99.6	88.8 / 97.4
MC3D-AD [8]	96.2 / –	80.5 / –	79.5 / –	70.9 / –	75.6 / –	93.0 / –	**97.8** / –	71.9 / –	88.5 / –	91.1 / –	75.4 / –
PLANE [38]	90.5 / 73.4	91.4 / 90.4	84.3 / 83.3	81.4 / 90.0	99.4 / 94.9	96.3 / 81.0	90.7 / 87.4	95.6 / 83.4	70.6 / 88.0	89.3 / 85.3	80.0 / 79.3
BTP† [18]	75.2 / 85.6	53.8 / 94.6	75.8 / 95.7	62.7 / 87.0	79.2 / 95.6	92.1 / 94.0	63.8 / 80.3	76.1 / 90.1	59.7 / 80.6	44.5 / 81.3	78.6 / 88.8
Ours	**100.0**$_{\pm 0.0}$ / **97.5**$_{\pm 0.2}$	**100.0**$_{\pm 0.0}$ / **99.6**$_{\pm 0.1}$	**100.0**$_{\pm 0.0}$ / **99.8**$_{\pm 0.1}$	**100.0**$_{\pm 0.0}$ / 89.3$_{\pm 0.3}$	**100.0**$_{\pm 0.0}$ / **97.2**$_{\pm 0.4}$	**100.0**$_{\pm 0.0}$ / **99.0**$_{\pm 0.2}$	96.4$_{\pm 0.2}$ / 93.1$_{\pm 0.4}$	**100.0**$_{\pm 0.0}$ / 92.6$_{\pm 0.3}$	**100.0**$_{\pm 0.0}$ / 95.5$_{\pm 0.2}$	**99.3**$_{\pm 0.1}$ / **99.9**$_{\pm 0.1}$	**93.4**$_{\pm 0.3}$ / **98.7**$_{\pm 0.1}$

Method	bucket0	bucket1	cap0	cap3	cap4	cap5	cup0	cup1	eraser0	headset0	headset1
BTF (RAW) [15]	61.7 / 61.7	32.1 / 68.6	66.8 / 52.4	52.7 / 68.7	46.8 / 46.9	37.3 / 37.3	40.3 / 63.2	52.1 / 56.1	52.5 / 63.7	37.8 / 57.8	51.5 / 47.5
BTF (FPFH)	40.1 / 40.1	63.3 / 63.3	61.8 / 73.0	52.2 / 65.8	52.0 / 52.4	58.6 / 58.6	58.6 / 79.0	61.0 / 61.9	71.9 / 71.9	52.0 / 62.0	49.0 / 59.1
M3DM [40]	40.9 / 69.8	50.1 / 69.9	55.7 / 53.1	42.3 / 60.5	77.7 / 71.8	63.9 / 65.5	53.9 / 71.5	55.6 / 55.6	62.7 / 71.0	57.7 / 58.1	61.7 / 58.5
PatchCore (FPFH) [33]	46.9 / 45.9	55.1 / 57.1	58.0 / 47.2	45.3 / 65.3	75.7 / 59.5	79.0 / 79.5	60.0 / 65.5	58.6 / 59.6	65.7 / 81.0	58.3 / 58.3	63.7 / 46.4
PatchCore (PMAE)	59.3 / 58.6	56.1 / 57.4	58.9 / 54.4	47.6 / 48.8	72.7 / 72.5	53.8 / 54.5	61.0 / 51.0	55.6 / 85.6	67.7 / 37.8	59.1 / 57.5	62.7 / 42.3
CPMF [6]	48.2 / 48.6	60.1 / 60.1	60.1 / 60.1	55.1 / 55.1	55.3 / 55.3	69.7 / 55.1	49.7 / 49.7	49.9 / 50.9	68.9 / 68.9	64.3 / 69.9	45.8 / 45.8
Reg3D-AD [23]	61.0 / 61.9	75.2 / 75.2	69.3 / 63.2	72.5 / 71.8	64.3 / 81.5	46.7 / 46.7	51.0 / 68.5	53.8 / 69.8	34.3 / 75.5	53.7 / 58.0	61.0 / 62.6
Group3AD [51]	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –
IMRNet [20]	58.0 / 58.5	77.1 / 77.4	73.7 / 71.5	77.5 / 70.6	65.2 / 75.3	65.2 / 74.2	64.3 / 64.3	75.7 / 68.8	54.8 / 54.8	72.0 / 70.5	67.6 / 47.6
R3D-AD [50]	68.3 / –	75.6 / –	82.2 / –	73.0 / –	68.1 / –	67.0 / –	77.6 / –	75.7 / –	89.0 / –	73.8 / –	79.5 / –
PointAD† [49]	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –
ISMP [22]	– / 52.4	– / 67.2	– / 86.5	– / 73.4	– / 75.3	– / 67.8	– / 86.9	– / 60.0	– / 70.6	– / 58.0	– / 70.2
PO3AD [43]	85.3 / 75.5	78.7 / 89.9	87.7 / 95.7	85.9 / 94.8	79.2 / 94.0	67.0 / 86.4	87.1 / 90.9	83.3 / **93.2**	99.5 / 97.4	80.8 / 82.3	92.3 / 90.7
RPRIL [45]	81.3 / 61.0	90.2 / 85.5	65.9 / 86.1	86.3 / 94.5	68.1 / 86.4	90.2 / 97.0	73.3 / 79.8	93.3 / 88.1	**100.0** / 98.0	**100.0** / **94.6**	84.3 / **97.0**
PASDF [47]	96.8 / 87.5	77.5 / 82.4	85.2 / 94.8	64.9 / 86.1	64.6 / 89.4	85.3 / 92.0	97.1 / 94.8	85.7 / 88.4	95.2 / 94.5	**100.0** / 86.3	79.5 / 89.1
CASL [46]	99.0 / 79.6	91.4 / 92.7	**94.8** / **97.4**	82.1 / **97.1**	77.2 / 94.9	55.4 / 66.9	**99.0** / **98.0**	66.5 / 79.7	99.5 / 87.6	83.6 / 77.0	**96.7** / 90.9
MC3D-AD [8]	89.8 / –	78.4 / –	79.3 / –	70.1 / –	**83.5** / –	76.1 / –	74.3 / –	**95.2** / –	77.6 / –	86.2 / –	88.6 / –
PLANE [38]	98.1 / 85.5	**96.8** / **94.7**	94.4 / 90.4	**95.4** / 95.1	73.0 / 89.4	87.0 / 84.4	80.5 / 83.4	70.5 / 84.7	**100.0** / 96.7	78.2 / 68.9	77.6 / 70.4
BTP† [18]	61.9 / 83.5	79.1 / 94.0	57.3 / 94.1	60.5 / 82.8	65.6 / 92.6	62.7 / 95.8	71.5 / 91.5	51.8 / 80.4	85.1 / 95.1	60.1 / 83.5	57.2 / 80.4
Ours	**99.5**$_{\pm 0.1}$ / **90.3**$_{\pm 0.2}$	93.1$_{\pm 0.4}$ / 94.4$_{\pm 0.3}$	92.4$_{\pm 0.4}$ / 96.2$_{\pm 0.5}$	87.6$_{\pm 0.2}$ / 95.6$_{\pm 0.3}$	82.0$_{\pm 0.5}$ / **97.0**$_{\pm 0.2}$	**93.1**$_{\pm 0.3}$ / **98.4**$_{\pm 0.1}$	97.9$_{\pm 0.3}$ / 95.5$_{\pm 0.3}$	91.5$_{\pm 0.2}$ / 92.2$_{\pm 0.2}$	**100.0**$_{\pm 0.0}$ / **99.3**$_{\pm 0.2}$	**100.0**$_{\pm 0.0}$ / 93.0$_{\pm 0.3}$	94.4$_{\pm 0.3}$ / 95.2$_{\pm 0.4}$

low-weighted. This is well-aligned with the design of $\beta_{i,\mathbf{x}}$: since it aggregates the `[CLS]`-to-patch attention across all transformer layers and heads, $\beta_{i,\mathbf{x}}$ is particularly sensitive to the geometric properties (e.g., curvature, boundary proximity) of the 3D regions. In addition, this aligns with findings in the 2D domain, where self-attention in Vision Transformers is known to focus on salient structural features [7, 9]. In the context of anomaly detection, this geometric sensitivity is critical as defects and anomalies in industrial objects often occur at geometrically complex regions such as edges, joints, and thin structures [1], and assigning higher importance to these regions allows the framework to be more sensitive to subtle deviations in the most vulnerable locations.

$\mu_{i,\mathbf{x}}^{(\mathbf{z})}$ and $\mu_{i,\mathbf{x}}^{(\mathbf{v})}$ capture 3D semantics at different granularities. The middle and bottom rows of Figure 11 display the linguistic grounding weights $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$ and $\mu_{i,\mathbf{x}}^{(\mathbf{v})}$, respectively. Both weights measure how well each patch-level representation aligns with the category text description in the joint 3D-language space, but they are derived from different feature sources: $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$ is computed from the post-transformer patch embedding $\mathbf{z}_{i,\mathbf{x}}^{L}$, while $\mu_{i,\mathbf{x}}^{(\mathbf{v})}$ is computed from the aggregated native pre-max-pooling point features $\tilde{\mathbf{v}}_{i,\mathbf{x}}$. As shown in Figure 11, $\mu_{i,\mathbf{x}}^{(\mathbf{v})}$ exhibits richer and more fine-grained weight variation across patches compared to $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$. In contrast, $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$ shows relatively more uniform distributions with less spatial variation in certain regions. This observation suggests that the channel-wise max-pooling in the tokenizer (Mini-PointNet) compresses $K{=}32$ point features into a single patch embedding, discarding fine-grained geometric details. As a result, $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$, being derived from the post-bottleneck representation, reflects a coarser view of linguistic grounding. In

Table 9: Comparison of O-AUROC and P-AUROC results (↑) of competing methods on Anomaly-ShapeNet (continued). Conventions as in the first part of this table.

Method	helmet0	helmet1	helmet2	helmet3	jar0	phone	shelf0	tap0	tap1	vase0	vase1
BTF (RAW) [15]	55.3 / 50.4	34.9 / 44.9	60.2 / 60.5	52.6 / 70.0	42.0 / 42.3	56.3 / 58.3	16.4 / 46.4	52.5 / 52.7	57.3 / 56.4	53.1 / 61.8	54.9 / 54.9
BTF (FPFH)	57.1 / 57.5	71.9 / 74.9	54.2 / 64.3	44.4 / 72.4	42.4 / 42.7	67.1 / 67.5	60.9 / 61.9	56.0 / 56.8	54.6 / 59.6	34.2 / 64.2	21.9 / 61.9
M3DM [40]	52.6 / 59.9	42.7 / 42.7	62.3 / 62.3	37.4 / 65.5	44.1 / 54.1	35.7 / 35.8	56.4 / 55.4	75.4 / 65.4	73.9 / 71.2	42.3 / 38.8	42.7 / 60.2
PatchCore (FPFH) [33]	54.6 / 54.8	48.4 / 48.9	42.5 / 45.5	40.4 / 73.7	47.2 / 47.8	38.8 / 48.8	49.4 / 61.3	75.3 / 73.3	76.6 / 76.8	45.5 / 65.5	42.3 / 45.3
PatchCore (PMAE)	55.6 / 58.0	55.2 / 56.2	44.7 / 65.1	42.4 / 61.5	48.3 / 48.7	48.8 / 88.6	52.3 / 54.3	45.8 / 85.8	53.8 / 54.1	44.7 / 67.7	55.2 / 55.1
CPMF [6]	55.5 / 55.5	58.9 / 54.2	46.2 / 51.5	52.0 / 52.0	61.0 / 61.1	50.9 / 54.5	68.5 / 78.3	35.9 / 45.8	69.7 / 65.7	45.1 / 45.8	34.5 / 48.6
Reg3D-AD [23]	60.0 / 60.0	38.1 / 62.4	61.4 / 82.5	36.7 / 62.0	59.2 / 59.9	41.4 / 59.9	68.8 / 68.8	67.6 / 58.9	64.1 / 74.1	53.3 / 54.8	70.2 / 60.2
Group3AD [51]	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –
IMRNet [20]	59.7 / 59.8	60.0 / 60.4	64.1 / 64.4	57.3 / 66.3	78.0 / 76.5	75.5 / 74.2	60.3 / 60.5	67.6 / 68.1	69.6 / 69.9	53.3 / 53.5	75.7 / 68.5
R3D-AD [50]	75.7 / –	72.0 / –	63.3 / –	70.7 / –	83.8 / –	76.2 / –	69.6 / –	73.6 / –	90.0 / –	78.8 / –	72.9 / –
PointAD† [49]	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –
ISMP [22]	– / 68.3	– / 62.2	– / 84.4	– / 72.2	– / 82.3	– / 66.1	– / 68.7	– / 52.2	– / 55.2	– / 61.1	– / 84.3
PO3AD [43]	76.2 / 87.8	96.1 / **94.8**	86.9 / 93.2	75.4 / 84.6	86.6 / 87.1	77.6 / 81.0	57.3 / 66.3	74.5 / 78.3	68.1 / 69.2	85.8 / 95.5	74.2 / 88.2
RPRIL [45]	81.7 / 92.5	98.6 / 90.6	87.5 / 89.1	87.6 / 95.6	**100.0** / 98.2	**100.0** / 99.2	57.7 / 63.2	**94.8** / **91.8**	80.4 / 86.9	99.6 / **98.0**	60.5 / 70.5
PASDF [47]	81.2 / 81.6	93.8 / 64.6	76.5 / 80.9	84.6 / 95.8	**100.0** / 95.9	**100.0** / 95.1	71.3 / 86.5	88.2 / 88.4	79.3 / 90.2	**100.0** / 94.4	**92.9** / 79.7
CASL [46]	78.3 / 92.3	89.0 / 88.2	88.1 / 92.9	86.1 / 98.0	99.5 / 97.4	98.1 / 83.8	83.2 / **92.8**	76.1 / 65.2	56.3 / 60.2	83.8 / 89.4	89.0 / **97.2**
MC3D-AD [8]	67.2 / –	**100.0** / –	60.9 / –	**97.9** / –	97.1 / –	91.9 / –	**84.1** / –	94.5 / –	**97.0** / –	82.1 / –	85.7 / –
PLANE [38]	70.4 / 77.9	54.3 / 62.1	**100.0** / 92.3	72.1 / 79.2	**100.0** / 91.1	**100.0** / 92.1	75.9 / 88.1	46.7 / 44.1	65.2 / 47.4	89.6 / 84.7	77.1 / 65.6
BTP† [18]	50.4 / 88.1	48.6 / 71.9	64.3 / 85.6	51.8 / 80.0	77.9 / 87.6	93.4 / 94.0	67.2 / 84.7	43.0 / 83.1	45.4 / 84.2	93.4 / 93.9	55.3 / 80.6
Ours	**83.2**$_{\pm 0.4}$ / **95.4**$_{\pm 0.2}$	99.8$_{\pm 0.2}$ / 92.9$_{\pm 0.3}$	90.6$_{\pm 0.3}$ / **95.8**$_{\pm 0.3}$	88.9$_{\pm 0.3}$ / **99.2**$_{\pm 0.2}$	**100.0**$_{\pm 0.0}$ / **99.4**$_{\pm 0.3}$	**100.0**$_{\pm 0.0}$ / **99.6**$_{\pm 0.1}$	81.5$_{\pm 0.5}$ / 90.1$_{\pm 0.2}$	93.1$_{\pm 0.4}$ / 90.0$_{\pm 0.2}$	85.2$_{\pm 0.3}$ / **92.7**$_{\pm 0.3}$	**100.0**$_{\pm 0.0}$ / 96.9$_{\pm 0.5}$	90.5$_{\pm 0.3}$ / 91.1$_{\pm 0.4}$

Method	vase2	vase3	vase4	vase5	vase7	vase8	vase9	*Average*
BTF (RAW) [15]	41.0 / 40.3	71.7 / 60.2	42.5 / 61.3	58.5 / 58.5	44.8 / 57.8	42.4 / 55.0	56.4 / 56.4	49.3 / 55.0
BTF (FPFH)	54.6 / 64.6	69.9 / 69.9	51.0 / 71.0	40.9 / 42.9	51.8 / 54.0	66.8 / 66.2	26.8 / 56.8	52.8 / 62.8
M3DM [40]	73.7 / 73.7	43.9 / 65.8	47.6 / 65.5	31.7 / 64.2	65.7 / 51.7	66.3 / 55.1	66.3 / 66.3	55.4 / 61.0
PatchCore (FPFH) [33]	72.1 / 72.1	44.9 / 43.0	50.6 / 50.5	41.7 / 44.7	69.3 / 69.3	66.2 / 57.5	66.0 / 66.3	56.8 / 58.0
PatchCore (PMAE)	74.1 / 74.2	46.0 / 46.5	51.6 / 52.3	57.9 / 57.2	65.0 / 65.1	66.3 / 36.4	62.9 / 42.3	56.2 / 57.7
CPMF [6]	58.2 / 58.2	58.2 / 58.2	51.4 / 51.4	61.8 / 65.1	39.7 / 50.4	52.9 / 52.9	60.9 / 54.5	55.9 / 57.3
Reg3D-AD [23]	60.5 / 40.5	65.0 / 51.1	50.0 / 75.5	52.0 / 62.4	46.2 / 88.1	62.0 / 81.1	59.4 / 69.4	57.2 / 66.8
Group3AD [51]	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –
IMRNet [20]	61.4 / 61.4	70.0 / 40.1	52.4 / 52.4	67.6 / 68.2	63.5 / 59.3	63.0 / 63.5	59.4 / 69.1	65.9 / 65.0
R3D-AD [50]	75.2 / –	74.2 / –	63.0 / –	75.7 / –	77.1 / –	72.1 / –	71.8 / –	74.9 / –
PointAD† [49]	– / –	– / –	– / –	– / –	– / –	– / –	– / –	– / –
ISMP [22]	– / 73.3	– / 76.2	– / 54.5	– / 47.2	– / 70.1	– / 85.1	– / 61.5	– / 68.9
PO3AD [43]	95.2 / 97.8	82.1 / 88.4	67.5 / 90.2	85.2 / 93.7	96.6 / 98.2	73.9 / 95.0	83.0 / 95.2	83.9 / 89.9
RPRIL [45]	**100.0** / 99.7	84.5 / 84.4	81.8 / 92.7	**100.0** / 87.9	64.3 / 86.3	81.8 / 93.4	**87.3** / **97.1**	86.1 / 88.2
PASDF [47]	**100.0** / 95.6	80.6 / 86.8	91.2 / 89.9	**100.0** / 91.5	**100.0** / 95.9	92.4 / 90.9	83.6 / 86.3	90.7 / 89.6
CASL [46]	98.1 / 97.1	**93.0** / **93.1**	82.1 / 91.3	71.4 / 73.2	**100.0** / 99.7	93.3 / **98.9**	81.8 / 88.6	88.7 / 89.9
MC3D-AD [8]	92.9 / –	76.1 / –	87.6 / –	97.6 / –	93.8 / –	67.0 / –	73.6 / –	84.2 / 74.8
PLANE [38]	97.1 / 82.3	78.2 / 78.5	77.3 / 82.6	69.0 / 72.7	93.8 / 92.6	**96.4** / 95.9	59.2 / 76.6	83.6 / 82.1
BTP† [18]	82.3 / 95.7	46.2 / 81.2	61.2 / 83.8	59.5 / 81.6	76.8 / 86.2	70.3 / 90.7	55.9 / 85.1	65.2 / 87.3
Ours	**100.0**$_{\pm 0.0}$ / **99.9**$_{\pm 0.1}$	91.7$_{\pm 0.2}$ / 90.5$_{\pm 0.3}$	**92.9**$_{\pm 0.2}$ / **94.8**$_{\pm 0.2}$	**100.0**$_{\pm 0.0}$ / **95.5**$_{\pm 0.4}$	**100.0**$_{\pm 0.0}$ / **99.9**$_{\pm 0.1}$	96.2$_{\pm 0.3}$ / 96.6$_{\pm 0.4}$	86.1$_{\pm 0.5}$ / 95.7$_{\pm 0.3}$	**95.0**$_{\pm 0.2}$ / **95.6**$_{\pm 0.3}$

Table 10: Comparison with native 3D-VLM methods for 3D anomaly detection (O-/P-AUROC averaged across all classes, ↑). All three methods use a ULIP-2 backbone; PLANE and BTP results are taken from their papers. Best results are in **bold**.

Method	**Protocol**	**Real3D-AD**	**Anomaly-ShapeNet**
PLANE [38]	Unified multi-class, unsupervised (pseudo-anomaly augmentation)	77.7 / 73.6	83.6 / 82.1
BTP [18]	Zero-shot (one labeled auxiliary category; no target-category data)	61.4 / 84.5	65.2 / 87.3
FACETS (ours)	Unified multi-class, unsupervised (normal samples only)	**93.1**$_{\pm 0.2}$ / **94.7**$_{\pm 0.2}$	**95.0**$_{\pm 0.2}$ / **95.6**$_{\pm 0.3}$

contrast, the point feature path bypasses this bottleneck by extracting pre-max-pooling features and aggregating them through the MLP g^{MLP}, thereby retaining fine-grained geometric information that enables $\mu_{i,\mathbf{x}}^{(\mathbf{v})}$ to capture more comprehensive and spatially detailed patterns.

Complementary roles of β and μ in the enhanced embedding. The visualization also reveals that $\beta_{i,\mathbf{x}}$ and the two μ weights highlight complementary aspects of each object. While $\beta_{i,\mathbf{x}}$ focuses on regions with distinctive 3D geometric structure (structural saliency), $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$ and $\mu_{i,\mathbf{x}}^{(\mathbf{v})}$ focus on regions whose features are well-aligned with the category's textual description (semantic relevance). These two types of importance do not always coincide, as a geometrically complex region may or may not be semantically distinctive for the category, and vice versa. This complementary behavior is also analogous to findings in 2D CLIP-based methods, where intra-image attention and text-visual alignment weights serve distinct but synergistic roles in capturing fine-grained visual details [37, 27].

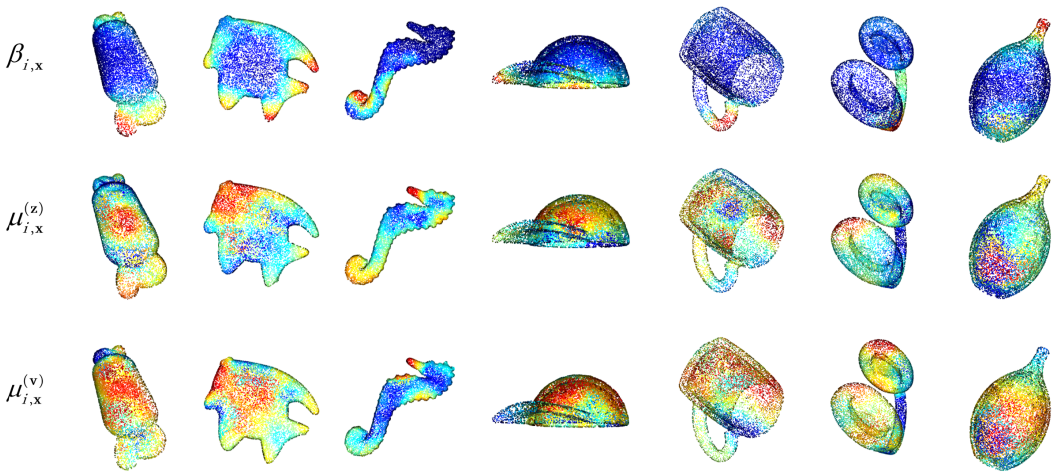


Figure 11: Visualization of our weight distributions on normal samples from Real3D-AD and Anomaly-ShapeNet datasets. In each row, we display the visualization of our proposed $\beta_{i,\mathbf{x}}$, $\mu_{i,\mathbf{x}}^{(\mathbf{z})}$, and $\mu_{i,\mathbf{x}}^{(\mathbf{v})}$ weight coefficients from top to bottom, respectively. Warm colors indicate high weight values and cool colors indicate low values.

G.3 Efficiency and Scalability Analysis

We compare the per-model cost of FACETS with recent state-of-the-art methods of all major paradigms: memory-bank-based ISMP [22], registration- and memory-bank-based RPRIL [45], reconstruction-based PO3AD [43] and CASL [46], and the unified multi-category MC3D-AD [8], which we run with its official code under the identical protocol. For PLANE [38], whose code was not publicly available at the time of writing, we report its published results only. All measurements use a workstation with a single NVIDIA RTX 3090 GPU at batch size 1 under identical software environments (Table 11). The table also lists each method's average AUROC, so that cost can be read against accuracy.

Parameters. FACETS has 86.4M parameters in total, more than the per-category models (10.2–29.4M) and MC3D-AD (34.2M). Of these, 85.7M are frozen (Point-BERT: 21.9M, text encoder: 63.4M, frozen projection matrix P: 0.39M), and only 0.76M are trainable: the shared learnable projection layer $\hat{P}$ (197K), the MLP g^{MLP} (247K), and the GNN (weight matrix W_{GNN}: 319K; edge encoding f_{edge}: 256). Because the category prompts are fixed, their text embeddings can be pre-computed and cached (a single $\bar{d}$-dimensional vector per category), which removes the 63.4M text encoder at inference and leaves a 23.0M inference-resident footprint.

Latency and memory. Despite its larger parameter count, FACETS has the lowest per-sample latency (210 ms) and peak GPU memory (2.1 GB) in the comparison, and the fewest GMACs among the methods where they were measured. It avoids the two most computationally expensive operations in 3D anomaly detection: (i) iterative registration (RPRIL; MC3D-AD, whose latency is

Table 11: **Efficiency and accuracy comparison** (one NVIDIA RTX 3090 GPU, batch size 1). Parameters are per model; the total of FACETS includes its frozen ULIP-2 backbone (85.7M), whose 63.4M text encoder can be dropped at inference by caching the fixed category text embeddings, leaving 23.0M parameters resident. GMACs are measured per sample with `fvcore` [26] (1 GMAC $= 10^9$ multiply-accumulate operations); peak GPU memory and latency are measured per sample at inference. AUROC is averaged across all classes. The last two columns aggregate over the C=12 categories of Real3D-AD (number of models required $\times$ total parameters per model; total training wall-clock time) and are secondary statistics. MC3D-AD is run with its official code; its latency is dominated by test-time registration. PLANE: published results only. Best results are in **bold**; "–" indicates results not available.

		Per model				O-/P-AUROC↑		All categories (secondary)	
Method	**Paradigm**	**Params (train. / total)**	**GMACs↓**	**Memory (GB)↓**	**Latency (ms)↓**	**Real3D-AD**	**Anomaly-ShapeNet**	**Deploy Params**	**Train Time (min)↓**
ISMP [22]	Mem. Bank	29.4M / 29.4M	18.6	6.8	2540	75.7 / 83.6	– / 68.9	352.8M	∼120
PO3AD [43]	Reconst.	10.8M / 10.8M	11.3	3.1	310	76.6 / –	83.9 / 89.9	129.6M	∼180
RPRIL [45]	Reg.+Bank	22.5M / 22.5M	28.1	7.3	3420	78.1 / 87.9	86.1 / 88.2	270.0M	∼150
CASL [46]	Reconst.	10.2M / **10.2M**	12.7	3.4	380	82.3 / 88.2	88.7 / 89.9	122.4M	∼200
MC3D-AD [8]	Reconst. (unified)	12.1M / 34.2M	–	5.5	3560	78.2 / 76.8	84.2 / 74.8	**34.2M**	–
PLANE [38]	VLM (unified)	–	–	–	–	77.7 / 73.6	83.6 / 82.1	–	–
Ours	VLM (unified)	**0.76M** / 86.4M	**10.3**	**2.1**	**210**	**93.1 / 94.7**	**95.0 / 95.6**	86.4M	**13.5**

dominated by test-time registration), which involves sequential nearest-neighbor searches and SVD computations that are inherently difficult to parallelize on GPUs; and (ii) k-nearest-neighbor search against stored memory banks (ISMP and RPRIL), with complexity $\mathcal{O}(N_{\text{test}} \times N_{\text{bank}} \times d)$, which also requires thousands of high-dimensional feature vectors to be kept in GPU memory. Instead, FACETS performs a single, fully parallelizable forward pass through the frozen 3D encoder backbone, followed by lightweight trainable modules, and needs no auxiliary feature storage. Its backbone therefore appears as a parameter and storage cost rather than as a runtime disadvantage.

All-category aggregates (secondary). Per-category methods train and store one model per category, so covering the 12 categories of Real3D-AD takes 12 models (122.4–352.8M parameters in total) and about 120–200 minutes of training, compared with one model and 13.5 minutes for FACETS, which computes gradients only for its 0.76M trainable parameters. These aggregates favor unified models by construction, since they assume that all categories are deployed together, so we report them only as secondary statistics; the per-model comparison above is the primary one.

G.4 Sensitivity Analysis for λ_{intra}

We analyze the sensitivity of our method to the intra-patch consistency loss weight λ_{intra}, which controls the strength of within-patch feature clustering regularization. We evaluate five values spanning nearly two orders of magnitude: $\lambda_{\text{intra}} \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$, and compare against the ablation baseline where $\mathcal{L}_{\text{intra}}$ is entirely removed ($\lambda_{\text{intra}} = 0$). Results are reported in Table 12. Note that λ_{intra} and other hyperparameters in our proposed method were selected on a pseudo-validation set constructed via Norm-AS-style synthetic anomalies [43] applied to training samples, and the test set was used only for the final evaluation. We report the sensitivity analysis results of λ_{intra} on the test set solely to characterize robustness, not for hyperparameter selection.

Across both datasets and metrics, all tested values of λ_{intra} outperform the ablation baseline without $\mathcal{L}_{\text{intra}}$. The improvements are particularly pronounced for point-level detection (P-AUROC), where even the least favorable setting ($\lambda_{\text{intra}} = 0.5$) yields gains of $+1.0$ on Real3D-AD and $+1.9$ on Anomaly-ShapeNet over the respective baselines. This demonstrates that intra-patch consistency regularization provides a robust benefit to the framework regardless of the specific weight chosen. In addition, both datasets exhibit a response curve peaking at $\lambda_{\text{intra}} = 0.1$ for both metrics before rounding. The optimal value of λ_{intra} is consistent across the two benchmarks, indicating that the balance between within-patch regularization and the discriminative contrastive objectives generalizes across datasets of different scales and characteristics.

Table 12: Sensitivity analysis of λ_{intra} on the Real3D-AD and Anomaly-ShapeNet datasets. The baseline without $\mathcal{L}_{\text{intra}}$ ($\lambda_{\text{intra}} = 0$) achieves 92.6/87.9 (O-AUROC/P-AUROC) on Real3D-AD and 94.5/90.4 on Anomaly-ShapeNet (see Tables 2 and 3). Best results are in **bold**.

Dataset	Metric	λ_{intra}				
		0.01	0.05	0.1	0.2	0.5
Real3D-AD	O-AUROC	92.8	93.0	**93.1**	93.0	92.7
	P-AUROC	90.8	93.1	**94.7**	92.6	88.9
Anomaly-ShapeNet	O-AUROC	94.8	95.0	**95.0**	94.9	94.7
	P-AUROC	93.5	94.7	**95.6**	94.2	92.3

Furthermore, object-level performance (O-AUROC) remains generally stable across the nonzero λ_{intra} values, varying by at most 0.4 on Real3D-AD and 0.3 on Anomaly-ShapeNet. In contrast, point-level performance (P-AUROC) exhibits considerably higher sensitivity, with a range of 5.8 on Real3D-AD and 3.3 on Anomaly-ShapeNet. This is because $\mathcal{L}_{\text{intra}}$ directly regularizes the within-patch point feature distribution that underlies the point-level anomaly scores, while the object-level score operates on the combined embedding. Also, at large λ_{intra} values (≥ 0.2), point-level performance degrades noticeably, particularly on Real3D-AD. Excessive intra-patch regularization forces point features toward uniformity within each patch, suppressing the fine-grained geometric variation necessary for distinguishing subtle point-level anomalies.

Empirical illustration of the trade-off effect with respect to $\mathcal{L}_{\text{intra}}$. To empirically illustrate the trade-off effect with respect to $\mathcal{L}_{\text{intra}}$, which is also predicted by Theorem 1, we measure the within-patch score statistics on the test sets of both benchmarks across all λ_{intra} values from Table 12 plus the baseline ($\lambda_{\text{intra}} = 0$). Theorem 1 is stated for a single anomalous point, where $\mathcal{N} = \{1, \ldots, K\} \setminus \{j^*\}$ is entirely non-anomalous. For empirical visualization on patches with multiple anomalous points, we instead let $\mathcal{A}$ be the anomalous set, let $\mathcal{N}$ be the non-anomalous set, and choose $j^* \in \arg\max_{a \in \mathcal{A}} s_a$. Therefore, the tail statistics are always computed over non-anomalous points only. In this multi-anomaly case, the exact union-bound factor is $|\mathcal{N}|$, which is upper-bounded by $K - 1$; using $K - 1$ in the visualization is therefore conservative. For each test sample and each patch containing at least one anomalous point j^*, we compute the within-patch standard deviation of anomaly score for non-anomalous points $\widehat{\sigma}_{\mathcal{N},\text{std}} = \left(\frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}} (s_j - \bar{s}_{\mathcal{N}})^2 \right)^{1/2}$, the anomaly separation $\Delta = s_{j^*} - \bar{s}_{\mathcal{N}}$, and their ratio $\Delta / \widehat{\sigma}_{\mathcal{N},\text{std}}$ (a descriptive empirical SNR proxy, see Appendix E.2 for justification of using $\widehat{\sigma}_{\mathcal{N},\text{std}}$ instead of $\widehat{\sigma}_{\mathcal{N},\text{tail}}$), averaged across all qualifying patches and all categories of each dataset.

Figure 12 shows that the empirical SNR proxy and the P-AUROC peak together at $\lambda_{\text{intra}} = 0.1$ on both benchmarks, which is well-aligned with the theorem's prediction where the ranking-based detection performance tracks the empirical SNR proxy. Since $\widehat{\sigma}_{\mathcal{N},\text{std}}$ is an empirical standard deviation rather than the theoretical tail scale $\sigma_{\mathcal{N},\text{tail}}$, this result should be interpreted as empirical alignment with the theorem's mechanism rather than as a direct numerical instantiation of the theorem's lower bound. We note that the empirical SNR proxy is computed for the local within-patch anomaly score (s_j^{local}) analyzed in Theorem 1, while P-AUROC is computed using the final combined point-level anomaly score ($s_j^{\text{point}} = s_j^{\text{local}} + s_j^{\text{consist}}$). The co-peaking behavior, therefore, supports the role of the local within-patch anomaly score component within the final combined point-level anomaly score, rather than constituting a direct bound on final P-AUROC. Figure 13 further decomposes the empirical SNR proxy into its two components, revealing the competing-forces dynamic discussed in Appendix E.1: $\widehat{\sigma}_{\mathcal{N},\text{std}}$ decreases monotonically as λ_{intra} increases (stronger within-patch consistency), while Δ remains roughly stable for small λ_{intra} but collapses sharply at $\lambda_{\text{intra}} \geq 0.2$ due to over-regularization of point features.

G.5 Sensitivity Analysis for Our Point-level Anomaly Score

In Sec. 3.9, we define the point-level anomaly score as $s_j^{\text{point}} = s_j^{\text{local}} + s_j^{\text{consist}}$, combining the within-patch deviation score and the point-patch consistency score with equal weights. Since both components share the same cosine scale, this equal-weight combination requires no balancing

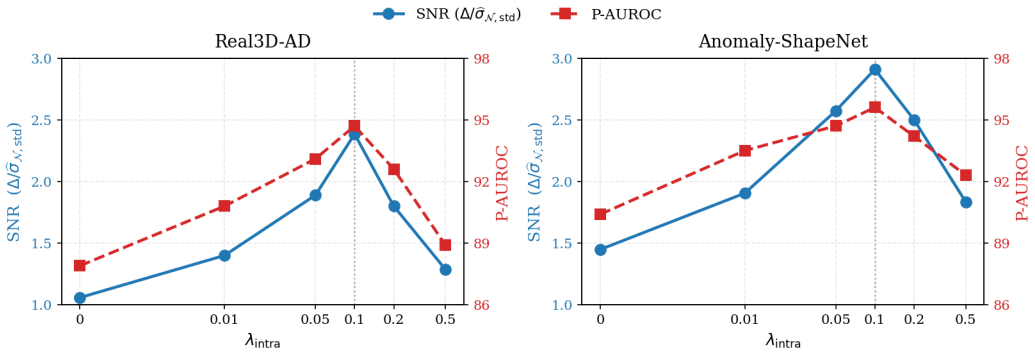


Figure 12: **Empirical illustration of the trade-off effect with respect to $\mathcal{L}_{\text{intra}}$.** The empirical SNR proxy ($\Delta/\widehat{\sigma}_{\mathcal{N},\text{std}}$, blue, left axis) and point-level AUROC (red, right axis) as a function of λ_{intra}, averaged across all categories of Real3D-AD (left) and Anomaly-ShapeNet (right). Both metrics co-peak at $\lambda_{\text{intra}} = 0.1$ (dotted vertical line), showing that the local rank-based detectability predicted by Theorem 1 is empirically aligned with the final P-AUROC trend.

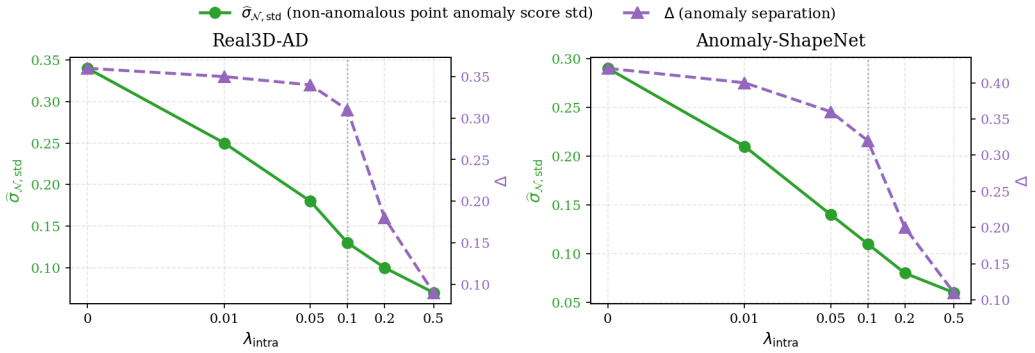


Figure 13: **Decomposition of the empirical SNR proxy into competing forces.** The empirical non-anomalous point anomaly score standard deviation $\widehat{\sigma}_{\mathcal{N},\text{std}}$ (green, left axis) and anomaly separation Δ (purple, right axis) as a function of λ_{intra}. As λ_{intra} increases, $\widehat{\sigma}_{\mathcal{N},\text{std}}$ decreases monotonically, while Δ initially remains stable and then collapses at large λ_{intra} due to over-regularization of point features. The optimal empirical trade-off is achieved at $\lambda_{\text{intra}} = 0.1$ (dotted vertical line).

hyperparameter. We conduct an *additional* sensitivity analysis to demonstrate the robustness of this choice. Specifically, we introduce a convex combination for the point-level anomaly score as:

$$s_j^{\text{point}} = (1 - \lambda_{\text{point}})\, s_j^{\text{local}} + \lambda_{\text{point}}\, s_j^{\text{consist}}, \quad \lambda_{\text{point}} \in [0, 1], \tag{37}$$

where $\lambda_{\text{point}} = 0$ corresponds to using s_j^{local} alone, $\lambda_{\text{point}} = 1$ corresponds to using s_j^{consist} alone, and $\lambda_{\text{point}} = 0.5$ is rank-equivalent to our default setup (since AUROC is invariant to positive scaling).

Note that all hyperparameters in our proposed method were selected on a pseudo-validation set constructed via Norm-AS-style synthetic anomalies [43] applied to training samples, and the test set was used only for the final evaluation. We report the sensitivity analysis results of λ_{point} on the test set solely to characterize robustness, not for hyperparameter selection.

The results are shown in Table 13. On both datasets, performance peaks at $\lambda_{\text{point}} = 0.5$. This reveals that the equal-weight setting, which is rank-equivalent to our default sum $s_j^{\text{point}} = s_j^{\text{local}} + s_j^{\text{consist}}$, achieves the best performance without additional tuning. Also, the peak P-AUROC at $\lambda_{\text{point}} = 0.5$ substantially exceeds the better of the two individual components on both datasets, indicating that the two anomaly-score components provide complementary anomaly evidence and that neither alone is sufficient. In addition, the performance remains within 1.0 point of the peak across the wide range

Table 13: Sensitivity analysis of the point-level anomaly score combination on Real3D-AD and Anomaly-ShapeNet (averaged P-AUROC across all classes, ↑). The two components are combined as $s_j^{\text{point}} = (1 - \lambda_{\text{point}})\, s_j^{\text{local}} + \lambda_{\text{point}}\, s_j^{\text{consist}}$.

Dataset	λ_{point}						
	0.0	0.1	0.3	0.5	0.7	0.9	1.0
Real3D-AD	91.8	92.9	94.3	**94.7**	93.8	91.4	90.3
Anomaly-ShapeNet	92.5	93.5	95.1	**95.6**	94.8	93.1	92.0

$\lambda_{\text{point}} \in [0.3, 0.7]$ on both datasets, indicating that our proposed method is robust to the exact choice of combination weight. This robustness further supports the equal-weight design as a principled, hyperparameter-free choice that does not require dataset-specific tuning.

G.6 Robustness to Noisy Point Clouds

We evaluate the robustness of our method to noisy point clouds on the Real3D-AD dataset under two settings, following the evaluation protocol established in RPRIL [45]. In **Setting 1**, Gaussian noise with a fixed standard deviation of $\sigma_{\text{noise}} = 0.005$ is injected into point clouds during training as a form of data augmentation, while varying noise levels ($\sigma_{\text{noise}} \in \{0, 0.001, 0.003, 0.005\}$) are applied at test time. In **Setting 2**, the model is trained exclusively on clean point clouds, and Gaussian noise is introduced only during inference.

Table 14: Robustness evaluation on noisy point clouds (Real3D-AD dataset). **Setting 1**: noise augmentation (σ_{noise}=0.005) applied during training; **Setting 2**: noise injected only at inference. Best result per (setting, metric, σ_{noise}) cell is in **bold**. RPRIL results are taken from [45].

Setting	**Method**	**Metric**	**Std (σ_{noise}) of injected Gaussian noise at inference**			
			0 (clean)	0.001	0.003	0.005
Setting 1 (train + inference)	RPRIL [45]	O-AUROC	78.1	76.7	77.7	76.8
		P-AUROC	87.9	87.5	86.4	83.1
	Ours	O-AUROC	**93.1**	**93.0**	**92.5**	**91.4**
		P-AUROC	**94.7**	**94.5**	**93.4**	**91.3**
Setting 2 (inference only)	RPRIL [45]	O-AUROC	78.1	76.9	63.0	56.6
		P-AUROC	87.9	86.5	69.1	56.5
	Ours	O-AUROC	**93.1**	**92.9**	**85.6**	**77.2**
		P-AUROC	**94.7**	**93.9**	**84.4**	**72.1**

As shown in Table 14, our method exhibits strong robustness to input perturbations under both settings. Under Setting 1, where noise augmentation is incorporated during training, performance degrades gracefully, demonstrating that mild noise augmentation effectively regularizes the model against test-time perturbations. Beyond robustness, this regularization mechanism is mechanistically aligned with our $\mathcal{L}_{\text{intra}}$ objective: training-time coordinate perturbations and $\mathcal{L}_{\text{intra}}$ both pressure the model to map nearby points within a patch to similar features, thereby reducing the empirical within-patch non-anomalous-score dispersion $\widehat{\sigma}_{\mathcal{N},\text{std}}$, which serves as a descriptive proxy for the upper-tail scale $\sigma_{\mathcal{N},\text{tail}}$ appearing in Theorem 1. Under Setting 2, the model trained on clean data experiences more pronounced degradation at higher noise levels ($\sigma_{\text{noise}} \geq 0.003$). The steeper decline in P-AUROC compared to O-AUROC is because point-level scoring is inherently more sensitive to local coordinate perturbations than object-level detection. Nevertheless, even in this more challenging setting without noise-aware training, our method consistently outperforms RPRIL [45] under the same protocol (Table 14).

G.7 Controlled Ablations

Table 15 reports controlled ablations that isolate the contribution of the FACETS modules over the ULIP-2 backbone, the role of language, and further design choices. Every variant is trained and evaluated with the protocol of the main results (mean and SEM over 10 random seeds).

Direct ULIP-2 baseline. This baseline uses the same frozen ULIP-2, prompts, and inference protocol as FACETS, but none of its modules. Its object-level anomaly score is the negative cosine similarity between the standard ULIP-2 representation and the category text embedding. At the point level, which vanilla ULIP-2 lacks, each point is scored by the negative patch-to-text cosine similarity of the patches that contain it (computed with the pooled-component block of the frozen P), averaged over those patches. It stays far below FACETS on both datasets and metrics: ULIP-2 supplies a shared 3D–text space but little anomaly detection capacity on its own, so the gap measures the contribution of the FACETS modules and losses.

Language contribution. Three variants isolate the text pathway. (i) *Learnable category embeddings* replace the category text embedding by a learnable vector $\mathbf{e}_c \in \mathbb{R}^{\bar{d}}$, which keeps category conditioning without text semantics. (ii) *Geometry-only* removes $\mu^{(\mathbf{z})}$, $\mu^{(\mathbf{v})}$, and both 3D–text losses, and scores each object by its distance to the per-category centroid of the training embeddings. (iii) *Shuffled prompts* apply a fixed wrong permutation $t_c \mapsto t_{\pi(c)}$ with $\pi(c) \neq c$ at both training and test time, which keeps conditioning but destroys its semantics. Geometry-only drops the most; learnable category embeddings recover part of the benefit of category conditioning but stay consistently below FACETS, and shuffled prompts fall in between. Semantically matched text anchors thus add value beyond category-index conditioning: the language pathway contributes a meaningful but incremental gain over learnable category conditioning, consistent with its role as category-level 3D–text grounding.

Prompt robustness. FACETS uses the template "A point cloud model of `[category]`". We also train and test FACETS with each of five alternative single templates, "a 3D scan of a `[category]`", "a geometric model of a `[category]`", "a point cloud representing a `[category]`", "a 3D object of category `[category]`", and "a CAD model of a `[category]`" (Table 15 reports their mean), and with a prompt ensemble that averages the text embeddings of the default template and the first four alternatives, at both training and test time. Both stay within 0.7 points of the default template on every metric, far less than the drop caused by shuffled or removed semantics, so varying the category-level wording changes little.

Design choices. *Separate* β: the patch embeddings keep the attention-derived $\beta_{i,\mathbf{x}}$, while the consolidated point representations are weighted by an independently learned set of weights $\beta^{(\mathbf{v})}_{i,\mathbf{x}}$. This variant is consistently worse than the shared $\beta_{i,\mathbf{x}}$, in line with Proposition 1: $\beta_{i,\mathbf{x}}$ encodes region-level saliency, a property of the 3D region rather than of a particular feature representation. *Normalized* w: replacing $w_{j,i,\mathbf{x}}$ in Eq. (5) by $w_{j,i,\mathbf{x}} / \sum_{k=1}^{K} w_{k,i,\mathbf{x}}$ slightly decreases performance, by 0.4–0.6 points. We keep w unnormalized on purpose. Writing $\tilde{\mathbf{v}}_{i,\mathbf{x}} = S_{i,\mathbf{x}}\, \mathbf{m}_{i,\mathbf{x}}$, with $S_{i,\mathbf{x}} = \sum_{j=1}^{K} w_{j,i,\mathbf{x}} \leq K$ and $\mathbf{m}_{i,\mathbf{x}}$ the mean of the projected point features under the normalized weights, per-patch normalization would only remove the factor $S_{i,\mathbf{x}}$, which lets the branch magnitude reflect the aggregate parent-aligned point evidence. This scale is bounded, $\|\tilde{\mathbf{v}}_{i,\mathbf{x}}\|_2 \leq S_{i,\mathbf{x}} \max_j \|g^{\text{MLP}}(\mathbf{v}_{j,i,\mathbf{x}})\|_2$, and at inference $S_{i,\mathbf{x}}$ affects only the object-level pathway: $\mu^{(\mathbf{v})}_{i,\mathbf{x}}$ is invariant to a positive rescaling of $\tilde{\mathbf{v}}_{i,\mathbf{x}}$ ($\hat{P}$ is a bias-free linear map inside a cosine), and the point-level scores involve neither w nor $\tilde{\mathbf{v}}$. *Object-level score*: using the maximum point-level score $\max_j s^{\text{point}}_j$ as the object-level score lowers O-AUROC, since the maximum relies on a single point and is fragile to the right tail of the normal-point score distribution, whereas our object-level score aggregates evidence across all patches.

Without an externally supplied category. The object-level score of FACETS uses the provided category, and we do not claim open-world or category-free inference; its point-level score needs no category. A label-free, closed-set variant with no extra parameters scores each test sample against the best-matching seen category, $s^{\text{infer}}_{\text{obj}}(\mathbf{x}) = -\max_{c \in \mathcal{C}_{\text{seen}}} \cos\big(M_{\text{3D}}(\mathbf{x}; t_c),\, M_{\text{text}}(t_c)\big)$, where $M_{\text{3D}}(\mathbf{x}; t_c)$ is the 3D representation computed with the prompt t_c. The best-matching category is the correct one for $98.6 \pm 0.2\%$ (Real3D-AD) and $94.1 \pm 0.3\%$ (Anomaly-ShapeNet) of the test samples; O-AUROC decreases by only 0.4/0.6 points, still well above the previous best methods, and

P-AUROC is unchanged by design. This variant removes the externally supplied label only over the training categories; generalization to unseen categories remains a limitation (Appendix H).

Table 15: Controlled ablations (averaged AUROC across all classes, ↑; mean and SEM over 10 random seeds). The last row is our full model. "–": unchanged, since P-AUROC does not depend on the object-level score.

		Real3D-AD		Anomaly-ShapeNet	
Group	**Variant**	O-AUROC	P-AUROC	O-AUROC	P-AUROC
Backbone	Direct ULIP-2 (no FACETS modules)	53.8±0.4	54.4±0.5	55.2±0.3	55.7±0.6
Language	Learnable category embeddings	91.3±0.2	93.7±0.1	93.0±0.1	94.4±0.1
	Geometry-only	88.3±0.2	92.6±0.4	89.5±0.3	93.6±0.3
	Shuffled prompts	89.2±0.3	92.8±0.2	90.3±0.3	93.9±0.4
Prompt	Alternative single templates (mean of 5)	92.6±0.1	94.2±0.1	94.4±0.2	94.9±0.1
	Prompt ensemble	92.7±0.1	94.2±0.2	94.5±0.1	95.0±0.2
Design	Separate β (learned $\beta^{(\mathbf{v})}$ for $\tilde{\mathbf{v}}$)	92.2±0.2	94.2±0.1	94.2±0.1	95.1±0.1
	Normalized w	92.5±0.1	94.3±0.1	94.5±0.2	95.0±0.1
	$\max_j s_j^{\text{point}}$ as object-level score	89.4±0.3	–	92.5±0.2	–
Category	Label-free (best-matching seen category)	92.7±0.1	–	94.4±0.2	–
FACETS (ours)		**93.1**±0.2	**94.7**±0.2	**95.0**±0.2	**95.6**±0.3

G.8 Mechanism Diagnostics

Retained point features encode point-level geometry. We fit linear probes that regress per-point surface normals and local curvature from the retained point features, and compare them with a control that broadcasts the post-pooling patch embedding to every point of its patch (Table 16). On both datasets, the retained features roughly halve the mean angular error of the normals and more than double the R^2 of the curvature regression. The pre-max-pooling features thus carry point-level geometric information that the broadcast patch embedding loses, which, together with the largest object-level drop when the $\mathbf{v}$-branch is removed (Table 2), supports the role of point geometric retention in FACETS.

Table 16: Linear-probe analysis of point features on Real3D-AD / Anomaly-ShapeNet (mean and SEM over 10 random seeds).

Point features	**Normals: mean angular error (°, ↓)**	**Curvature: R^2 (↑)**
Patch embedding broadcast	38.6±0.3 / 32.5±0.3	0.27±0.01 / 0.36±0.02
Retained point features (ours)	**20.9**±0.1 / **15.7**±0.2	**0.62**±0.02 / **0.77**±0.02

Localized defects shift the global embedding. The object-level score compares a global embedding with the category text embedding, which assumes that localized defects change the global representation. We group the anomalous test samples by their fraction of anomalous points and report, for each group, the mean shift $1 - \cos\left(M_{3D}(\mathbf{x}), \bar{M}_{3D}^{(c)}\right)$ from the per-category centroid $\bar{M}_{3D}^{(c)}$ of the normal training embeddings, and the object-level detection rate at a false-positive rate of 0.1 (Table 17). For reference, normal test samples shift by 0.03 on both datasets. The shift and the detection rate grow monotonically with defect size, and even the smallest group exceeds the normal level, so localized defects measurably move the global representation. Small defects are, however, where the object-level score is weakest: the smallest Real3D-AD group has the lowest detection rate (73.5%), while point-level localization remains strong for it (P-AUROC 93.2 vs. 94.7 overall).

H Limitations and Future Work

While FACETS sets a new state of the art in average AUROC with a native 3D vision–language framework for 3D anomaly detection, we acknowledge several limitations that suggest promising directions for future work.

Table 17: Defect-size sensitivity of the object-level score. Anomalous test samples are grouped by their fraction of anomalous points. Shift: mean $1 - \cos(M_{3D}(\mathbf{x}), \bar{M}_{3D}^{(c)})$ to the per-category centroid of normal training embeddings (normal test samples: 0.03 on both datasets). Detection rate: object-level, at a false-positive rate of 0.1.

Dataset	Anomalous-point fraction	Mean embedding shift	Detection rate (%)
Real3D-AD	$< 2\%$	0.05	73.5
	$[2\%, 4\%]$	0.08	83.1
	$> 4\%$	0.12	91.0
Anomaly-ShapeNet	$[1\%, 3\%)$	0.06	82.2
	$[3\%, 5\%)$	0.10	88.6
	$[5\%, 10\%]$	0.16	94.3

Frozen backbone. FACETS builds on a frozen ULIP-2 backbone and therefore inherits the representational coverage of its pre-training corpus. As large-scale 3D pre-training continues to mature and broader 3D vision–language foundation models become available, which is analogous to the trajectory of CLIP in the 2D domain, the representational capacity available to our framework is expected to grow correspondingly, further amplifying the benefits of our design.

Category-level text. FACETS uses a simple, deterministic category-level prompt template ("A point cloud model of `[category]`") for category-level grounding, rather than reasoning over textual descriptions of anomaly patterns. Our controlled ablations (Appendix G.7) show that this grounding yields a meaningful but incremental gain over learnable category conditioning, and that alternative single templates and a multi-template ensemble perform on par with, and slightly below, the default template, so varying the category-level wording changes little. Richer, instance-level descriptions, e.g., captions of normal geometry, could provide finer linguistic grounding and are a direction for future work.

Known test category. FACETS assumes that the test category is known for standard object-level inference, as do the per-category baselines, PLANE [38], and MC3D-AD [8]; only the zero-shot PointAD [49] and BTP [18] do not (Appendix A.4 audits how each method uses category information). The point-level score of FACETS needs no category. A label-free variant that selects the best-matching seen category costs only 0.4/0.6 O-AUROC on Real3D-AD/Anomaly-ShapeNet (Appendix G.7), but it still assumes a closed set of training categories; open-world generalization to unseen categories remains future work.

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 main claims made in the abstract and introduction accurately reflect the paper's contributions and scope, including category-level 3D–text grounding and results averaged over classes; limitations are stated in Section 6 and Appendix H.

   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 limitations are summarized in the Conclusions (Section 6) and discussed in Appendix H.

   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: [Yes]

Justification: Section 4 and Theorem 1 state the assumptions (a preserved anomaly margin and an upper-tail concentration condition) and the scope (the within-patch component of the point-level score); the complete proof is provided in Appendix E.

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: The paper describes the comprehensive reproduction details in Appendix F.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The code is publicly available at `https://github.com/infinitr0us/FACETS`, with a README that gives the environment, the data preparation, and the commands for training and evaluation; a copy is also provided as supplementary material on OpenReview. Both datasets are publicly available benchmarks.

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: The paper presents comprehensive experimental details.

   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: [Yes]

   Justification: The paper presents the Standard Error of the Mean (SEM) based on 10 runs of different random seeds in experimental results.

   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 detailed information on the compute resources of experiments: the hardware in Appendix F, and per-model latency, peak memory, and training time in Appendix G.3.

   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 paper complies with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: In the Introduction, it is stated that 3D anomaly detection is an important mechanism to ensure quality and reliability in industrial manufacturing.

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 poses no such risks.

    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: The assets used in the paper are credited by citation and used under their terms: the ULIP-2 code (BSD-3-Clause license) and its released Objaverse-only pre-trained Point-BERT checkpoint [42] (Appendix F), and the Real3D-AD [23] and Anomaly-ShapeNet [20] datasets (CC BY 4.0 license).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The code is released publicly at the URL given in the abstract under the MIT license, with a README that documents the environment setup, data preparation, training, and evaluation.

    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 nor 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 crowdsourcing nor research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    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.