Neural Networks

PointCore: An Efficient Framework for Unsupervised Point Cloud Anomaly Detection Using Joint Local-Global Features

--Manuscript Draft--

Manuscript Number:	NEUNET-D-25-02957R3
Article Type:	Article
Section/Category:	Learning Systems
Keywords:	Point cloud anomaly detection; Unsupervised learning; Memory bank
Corresponding Author:	Baozhu Zhao South China University of Technology CHINA
First Author:	Baozhu Zhao
Order of Authors:	Baozhu Zhao
	Xiaohan Zhang
	Jingfeng Guo
	Qi Liu
Abstract:	The foundation for various applications, such as industrial inspection and autonomous driving, is built upon the detection of anomaly data points in a training set using three-dimensional point cloud anomaly detection. Nevertheless, current point cloud anomaly detection techniques frequently use multiple feature memory banks to retain both local and global representations completely, but this leads to increased computational complexity and feature mismatches. To tackle this issue, we propose PointCore, a novel framework that introduces a unified coordinate-semantic memory bank. This architecture leverages low-dimensional coordinates to guide the matching process in the high-dimensional semantic feature space, effectively mitigating feature mismatches inherent in previous methods and reducing computational overhead. In addition, a normalization ranking method has been implemented to protect against outliers by not only standardizing values of varying scales to a common scale but also converting closely grouped data into a consistent distribution. Extensive testing on the Real3D-AD dataset shows that PointCore delivers fast inference time and outperforms the Reg3D-AD approach and other competitors in both detection and localization.

Dear Editor,

Re: Submission of Revised Manuscript (Ref. No.: NEUNET-D-25-02957R2)

On behalf of the co-authors, I am submitting the revised manuscript titled "PointCore: An Efficient Framework for Unsupervised Point Cloud Anomaly Detection Using Joint Local-Global Features" for Neural Networks.

We thank Reviewer #1 for recommending acceptance. In this revision, we focused on addressing the comments from Reviewer #4 by:

Clarifying Experimental Settings: Explicitly stating the unsupervised nature of our method and confirming that parameters were not tuned on the test set to avoid overfitting.

Adding Theoretical Insights: Explaining the rationale behind our Unified Memory Bank in Section 3.2.

Updating Related Works: Adding a new comparison table (Table 1) and including the suggested references on broader deep learning applications.

Clarifying Statistics: Confirming the stability of our results with standard deviations.

A detailed "Response to Reviewers" and the marked manuscript are attached.

Thank you for your time and consideration.

Sincerely,
Baozhu Zhao

Response to Reviewers' Comments (Round 3)

We would like to express our sincere gratitude to the Reviewers for their insightful feedback, which has helped us further refine our manuscript. We have carefully addressed all remaining concerns, particularly regarding the theoretical justification of our unified memory bank, the clarification of experimental rigor, and the inclusion of a comprehensive related work comparison.
Below, we provide a point-by-point response to the comments. All changes in the revised manuscript are highlighted in red.

Response to Reviewer #1

> *I have reviewed the final adjustments made to the manuscript. As indicated in the previous round, the authors have satisfactorily addressed my concerns. I maintain my recommendation to accept the paper for publication.*

Our Response: We sincerely thank Reviewer #1 for the time and effort dedicated to reviewing our work and for the continued support and recommendation for acceptance.

Response to Reviewer #4

> *1. Parameters of network have been enhanced using training data "until the model obtains the maximum accuracy". If this accuracy is the training accuracy, maybe over-fitting has been performed. 2. If this accuracy is the testing accuracy, the system is adjusted over the same subset that is evaluated. A validation subset could be used to optimize the system with different data than the testing data and without performing over-fitting. In addition, it would be interesting to know which range of each parameter has been analyzed."?*

Our Response: We thank the reviewer for this important query regarding experimental rigor and potential overfitting. We would like to clarify that our proposed **PointCore is an unsupervised anomaly detection framework**.

1. **Unsupervised Nature:** Unlike supervised learning, our training phase involves only "normal" samples without anomaly labels. Therefore, there is no "training accuracy" to optimize during the training phase, and the model does not update weights based on loss calculation against ground truth labels.

2. **Parameter Selection & Fairness:** As detailed in **Table 6** and the revised **Section 4.1 (Implementation Details)**, our hyperparameters (such as coreset percentage 1%) were not tuned by "peeking" at the test set. Instead, to ensure a fair comparison and reproducibility, they were **intentionally aligned with the baseline method Reg3D-AD** without modification.

3. **Parameter Ranges:** Since our goal was to demonstrate the architectural advantage of the Unified Memory Bank over the Split Memory Bank (Reg3D-AD) under **identical settings**, we intentionally **did not perform a grid search** or analyze a wide range of parameters on the test set. This constraint was self-imposed to strictly prevent "p-hacking" or overfitting to the test data.

Action: We have added an explicit clarification in **Section 4.1** (highlighted in red) stating that: *"To rigorously avoid overfitting and data leakage, all key hyperparameters... were adopted directly from the baseline method Reg3D-AD without further tuning on the test set."*

> *3. The motivation for replacing multiple memory banks with a unified coordinate-semantic memory bank is practical and compelling. However, it would benefit from theoretical or empirical insights explaining why coordinate information is an effective supervisory signal for semantic matching.*

Our Response: We appreciate this suggestion to deepen the theoretical justification of our approach. The rationale for using coordinate information is based on two key insights specific to **rigid body industrial anomaly detection**:

1. **Resolving Feature Ambiguity (Theoretical Insight):** Deep semantic features (e.g., from PointMAE) often capture local geometric patterns but lack global context. In rigid industrial objects, spatially disparate parts may share similar local geometries (e.g., two flat surfaces at different locations). Without coordinate constraints, a traditional memory bank might incorrectly match an anomaly to a semantically similar but spatially distant normal point, leading to a false negative. Coordinate proximity acts as a **strong spatial prior**, resolving this ambiguity by enforcing that a point is compared only against its structurally relevant neighbors.

2. **Structural Consistency (Empirical Insight):** Empirically, this is validated by **Figure 4** in our manuscript. As shown in the ”Starfish” example, without coordinate guidance (baseline), the feature on one arm incorrectly matches a different arm. Our unified approach enforces structural consistency, ensuring anomalies are compared against the correct corresponding part of the prototype.

Action: We have revised **Section 3.2 (Rationale for the Unified Coordinate-Semantic Memory Bank)** to explicitly discuss the resolution of ”feature ambiguity” and ”structural consistency” as the theoretical motivation.

> *4. How are the coordinate and semantic vectors aligned during matching?*

Our Response: We apologize if the mechanism was not fully clear. We would like to clarify that our method **does not require explicit feature alignment** (e.g., projecting coordinates and semantic features into a shared latent space). Instead, the ”alignment” is **inherent and structural**:

- **Physical Correspondence:** In our unified memory bank, each coordinate vector M_{i_c} is physically tied to the semantic feature vector M_{i_p} extracted from the exact same point index. They are naturally ”aligned” by their physical origin.
- **Hierarchical Matching:** During inference, we do not strictly fuse vectors for the search itself. We use coordinates strictly for **retrieval** (finding the spatial neighborhood) and semantic features strictly for **scoring** (computing similarity).

Thus, no learnable alignment or projection is needed.
Action: We have added a clarification in **Section 3.3** (highlighted in red) to explicitly state that the alignment is established through structural correspondence rather than learnable projections.

> *5. I suggest summarizing the related works into a table with respect to their characteristics. The authors should put their proposal into this table for easy comparison. Furthermore, I have listed some related works that authors should consider for evaluation of their work. Add more recent works in related section such as Dynamic multi-graph spatio-temporal learning... [List of citations]*

Our Response: We thank the reviewer for this constructive suggestion.

1. **Comparison Table:** We have added a new **Table 1** in **Section 2 (Related Work)** that summarizes representative methods based on their Modality, Core Approach, Registration usage, and Key Characteristics. This table clearly highlights the differences between our Unified Memory Bank approach and previous Split Memory Bank methods (like Reg3D-AD).

2. **Suggested Citations:** We have carefully reviewed the suggested literature. We have included **”3D-MMFN: Multi-level multimodal fusion network for 3D industrial image anomaly detection”** in our related work section and table, as it is highly relevant to the domain of 3D industrial inspection.

   Regarding the other suggested works on traffic flow prediction, cloud computing, and power systems: We acknowledge the broader success of deep learning in these fields. To provide a comprehensive view of anomaly detection applications beyond industrial inspection, we have briefly mentioned these domains in the **Introduction** (Section 1), citing representative works such as *Ali et al. (2025a), Ali et al. (2025b), Sharafian et al. (2025), and Ahmed et al. (2022).*

Action: We added **Table 1** to summarize related works and integrated the relevant citations into the **Introduction** and **Related Work** sections.

> *6. Do the results shown in various figures refer to a single run or multiple runs (average)? In the latter case, I will suggest adding standard deviation bars. The reason behind this is to ensure that the results overlap with the closest rivals or not.*

Our Response: We fully agree on the importance of statistical stability.

- **Data Already Included:** We would like to direct the reviewer to **Table 8 (Inference Time and Stability)** (formerly Table 7). This table explicitly reports the **mean ± standard deviation** for O-AUROC and P-AUROC over multiple runs with different random seeds.
- **No Overlap Issue:** As shown in Table 8, PointCore demonstrates significantly higher stability (O-AUROC SD = **0.03**) compared to the baseline Reg3D-AD (SD = **0.29**). This large difference in variance confirms that our performance gains are statistically significant and distinct from the baseline.
- **Figures:** For the visual plots (e.g., Fig. 7 and Fig. 8), we plotted the **average** of these multiple runs to maintain visual clarity, as adding error bars to these specific line plots would result in significant visual clutter. The statistical quantification is fully provided in Table 8.

> *7. The results on Real3D-AD are promising, but an ablation study isolating each component (memory bank, ranking normalization, coordinate guidance) would demonstrate the contribution of each module more clearly.*

Our Response: We thank the reviewer for highlighting this. We would like to confirm that comprehensive ablation studies isolating these exact modules were already included and have been further highlighted in **Section 4.5**:

- **Coordinate Guidance (Unified Memory Bank): Table 13** (formerly Table 12) compares our full model against variants using ”PointMAE Only” (no coordinate guidance). The results show that removing coordinate guidance causes the O-AUROC to drop from **0.829** to **0.721**, proving its critical role.
- **Ranking Normalization & Local Optimization: Table 10** isolates the contribution of the Ranking Block (RB) and Local Optimization (LR). The data demonstrates that adding the Ranking Block significantly improves the O-AUROC, confirming its effectiveness.

> *8. Furthermore, the introduction section needs considerable effort (concise and brief). The problem being investigated should be described clearly, but before that, the field of research should be made clearer. Furthermore, briefly describe the major contributions in bullet form, just before the organization paragraph.*

Our Response: We have revised the **Introduction** to be more concise and focused.

- We have clarified the broader field of research by acknowledging anomaly detection applications in diverse domains (incorporating the suggested references) before narrowing down to the specific challenges of 3D point cloud inspection.
- The specific problem of ”feature mismatch” in existing dual-memory bank methods is clearly described.
- As requested, the **major contributions** are clearly listed in **bullet form** at the end of the Introduction section.

Highlights

PointCore: An Efficient Framework for Unsupervised Point Cloud Anomaly Detection Using Joint Local-Global Features

Baozhu Zhao, Xiaohan Zhang, Jingfeng Guo, Qi Liu

- Propose a novel architecture named PointCore, which uses a single memory bank to store local (coordinate) and global (PointMAE) features, reducing computational cost and feature mismatch interference.
- Introduce a ranking-based normalization method to eliminate the distribution differences of anomaly scores, and adopt the point-to-plane Iterative Closest Points algorithm to optimize the point cloud registration results, enhancing the robustness of decision-making.
- Experiments on the Real3D-AD dataset show that compared with methods like Reg3D-AD, PointCore achieves competitive inference time and the best performance in both detection and localization.
- Combine global-local registration and memory bank construction, comprehensively utilize point cloud information, improve the recall rate of anomaly detection and reduce the false positive rate, suitable for scenarios that require precise detection of defective samples.

LaTeX source files

Declaration of Interest Statement

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for this journal and was not involved in the editorial review or the decision to publish this article.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

PointCore: An Efficient Framework for Unsupervised Point Cloud Anomaly Detection Using Joint Local-Global Features

Baozhu Zhao[a], Xiaohan Zhang[a], Jingfeng Guo[a] and Qi Liu[a,*]

[a]*Department of Future Technology, South China University of Technology, Guangzhou 511400, China.*

ARTICLE INFO

Keywords:
Point cloud anomaly detection
Unsupervised learning
Memory bank

ABSTRACT

The foundation for various applications, such as industrial inspection and autonomous driving, is built upon the detection of anomaly data points in a training set using three-dimensional point cloud anomaly detection. Nevertheless, current point cloud anomaly detection techniques frequently use multiple feature memory banks to retain both local and global representations completely, but this leads to increased computational complexity and feature mismatches. To tackle this issue, we propose PointCore, a novel framework that introduces a unified coordinate-semantic memory bank. This architecture leverages low-dimensional coordinates to guide the matching process in the high-dimensional semantic feature space, effectively mitigating feature mismatches inherent in previous methods and reducing computational overhead. In addition, a normalization ranking method has been implemented to protect against outliers by not only standardizing values of varying scales to a common scale but also converting closely grouped data into a consistent distribution. Extensive testing on the Real3D-AD dataset shows that PointCore delivers fast inference time and outperforms the Reg3D-AD approach and other competitors in both detection and localization.

1. Introduction

Anomaly detection aims to find the abnormal region of products and plays an important role in various fields, such as industrial quality inspection Bergmann, Batzner, Fauser, Sattlegger and Steger (2021); Carrera, Manganini, Boracchi and Lanzarone (2017); Song and Yan (2013), and autonomous driving Blum, Sarlin, Nieto, Siegwart and Cadena (2019); Hendrycks, Basart, Mazeika, Mostajabi, Steinhardt and Song (2019). Beyond visual inspection, deep learning-based prediction and anomaly monitoring techniques have also demonstrated significant success in diverse domains, including intelligent transportation systems (Ali, Naeem, Sharafian, Qiu, Wu and Bai, 2025a; Ali, Ali, Naeem, Asad, Alsarhan and Heyat, 2024), edge and cloud computing resource allocation (Ali, Ullah, Singh, Jiang, Alturise and Bai, 2025b; Ali, Shen, Huang, Zhang and Ali, 2017), power system control (Sharafian, Naeem, Ullah, Ali, Qiu and Bai, 2025), and heterogeneous network architecture (Ahmed, Zakarya, Liu, Khan, Ali and Khan, 2022). Current anomaly detection methods Deng and Li (2022); Du, Gu, Qin, Xie, Wang and Hu (2025); Roth, Pemula, Zepeda, Scholkopf, Brox and Gehler (2022); Zavrtanik, Kristan and Skočaj (2021) are mostly unsupervised and they target on two-dimensional (2D) images, where models are typically trained on RGB or grayscale images with well-studied architectures. The task of anomaly detection based on 3D point clouds is still relatively unexplored in the current literature. 3D point clouds offer richer structural information compared to 2D images, but they also come with challenges such as disorder, high sparsity, and irregular distribution. To process point clouds distinctly, various handcrafted or deep learning-based feature descriptors at different scales are applied.

Real3D-AD, a large-scale dataset for high-resolution 3D anomaly detection, was recently introduced by Liu, Xie, Chen, Li, Wang, Liu, Wang and Zheng (2023). The items in the Real3D-AD dataset feature a resolution of 0.0010mm-0.0015mm, a 360-degree range, and flawless prototypes. In Liu et al. (2023), the authors applied PatchCore Roth et al. (2022) from image anomaly detection to point cloud anomaly detection, and developed a general-purposed registration-based point cloud anomaly detection method, dubbed as Reg3D-AD. Reg3D-AD uses a method of dual-feature representation to maintain the local and global features of training prototypes, resulting in impressive detection accuracy but slow inference speed. Current point cloud anomaly detection methods can be categorized into twofolds: (1) Reconstruction-based methods, which reconstruct the input point cloud data via auto-encoder Li and Xu (2023); Ma, Wu and Liu (2025); Masuda, Hachiuma, Fujii, Saito and Sekikawa (2021a) and identify anomalies by comparing the deviation between the original and reconstructed data. Nevertheless, these methods are impacted by the resolution of point clouds, resulting in decreased inference speed and lower accuracy. (2) Memory bank-based approaches Defard, Setkov, Loesch and Audigier (2020); Fan, Ge, Zhang, Wang, Wu and Wu (2024); Li, Wang, Wang and Chen (2025); Liu et al. (2023), where memory banks help store typical characteristics to create a standard distribution implicitly and identify any defects that fall outside of this distribution. Utilizing a pre-trained feature extractor directly for creating a memory bank results in quicker training speed and is not influenced by the resolution of point clouds, unlike the previous method. In addition, existing point cloud anomaly detectors Horwitz and Hoshen (2022); Liu et al. (2023) often separating geometric and semantic features into independent memory banks force a disjoint nearest-neighbor search. This often leads to mismatches

*Corresponding author: Qi Liu
✉ drliuqi@scut.edu.cn (Q. Liu)
ORCID(s): 0000-0001-5378-6404 (Q. Liu)

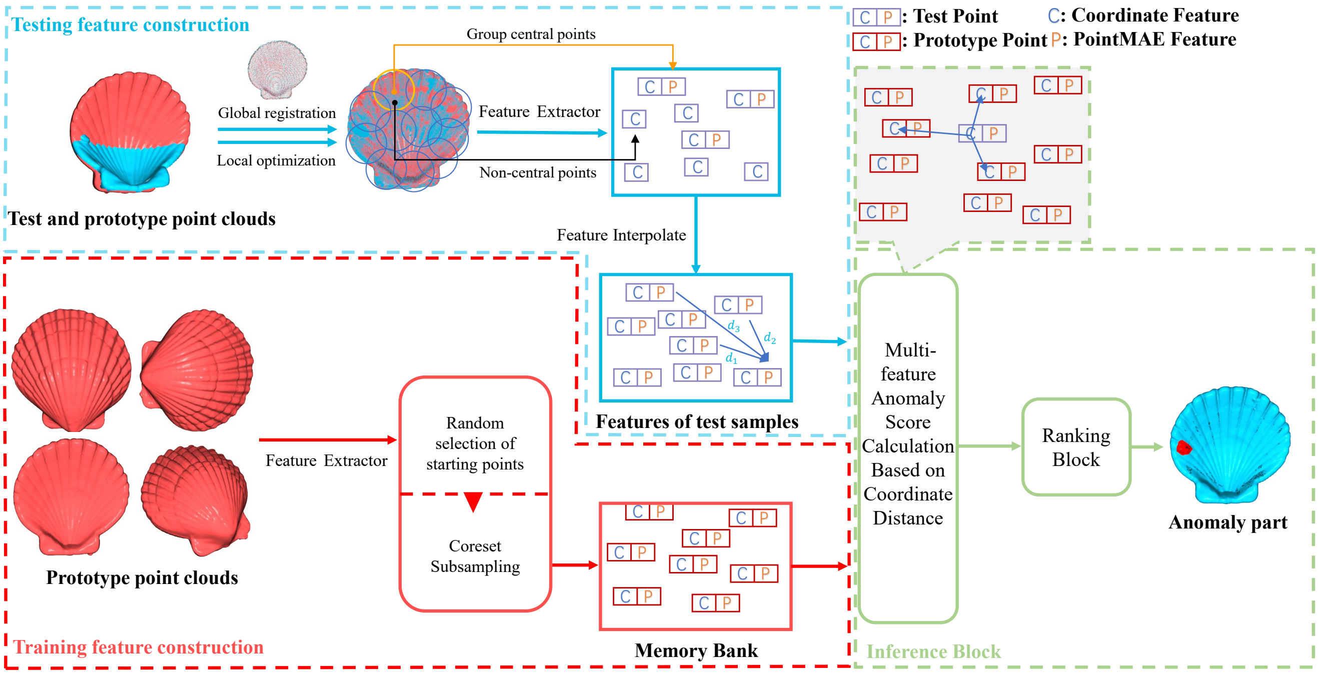


Figure 1: The pipeline of PointCore architecture. We chose a template point cloud at random to serve as the reference coordinate system, then used global registration and local optimization methods to convert other templates or test point clouds to match this reference coordinate system. After that, PointMAE pre-trained feature extractors are applied to get the PointMAE feature value of the chosen coordinates, connecting the coordinate value with the feature value, setting up the coordinate-PointMAE memory bank, and ultimately calculating anomaly scores for all points using the inference block.

where semantically similar but spatially distant patches are incorrectly paired, a critical issue in 3D anomaly detection where spatial consistency is paramount.

We propose using a PointCore framework for unsupervised anomaly detection in point clouds, incorporating both local and global features in collaboration to address the mentioned issues. In particular, our contributions can be summarized as follows:

1. We propose a novel unified memory bank architecture that tightly couples local coordinate information with global semantic features. Unlike prior works that treat them separately, our design uses coordinate proximity as a strong prior to constrain the search space for semantic matching, thus creating a more principled and effective feature retrieval mechanism for 3D data.

2. We introduce a ranking-based normalization method that proves crucial for effectively fusing anomaly scores from heterogeneous feature spaces (3D coordinates vs. 1154D embeddings), making the final decision more robust to outliers and distributional shifts.

3. Extensive testing on the Real3D-AD dataset shows that PointCore has competitive inference time and outperforms the state-of-the-art Reg3D-AD approach and other competitors in both detection and localization.

2. Related Work

2.1. 2D Anomaly Detection

Lately, there has been significant improvement in the detection accuracy, speed, and model stability of unsupervised anomaly detection for 2D data like RGB or grayscale images Ehret, Davy, Morel and Delbracio (2019); Pang, Shen, Cao and Hengel (2022a); Schwartz, Arbelle, Karlinsky, Harary, Scheidegger, Doveh and Giryes (2024); Xie, Wang, Liu, Lyu, Liu, Wang, Zheng and Jin (2023). Image-based anomaly detection methods can be categorized into image similarity-based and feature similarity-based. Image similarity-based models perform comparisons at the pixel level. Their goal is to create an image that closely resembles the original sample, with changes only made to the damaged areas. As a result, the difference map between the created and original images can indicate the probability of a defect being present, determining if the entire image has any abnormalities. The segmented results of defects are obtained through thresholding. The autoencoder and generative models are commonly used, such as autoencoders (AE) Hu, Li, Luo, Karimi and Zhang (2025); Mei, Wang and Wen (2018); Wang, Gu, Gu and Hu (2024), variational autoencoders (VAE) Fan, Wen, Li, Qiu, Levine and Xiao (2020); Lu, Wang, Shen, Shao and Tang (2023), and generative adversarial networks (GAN) Chen, Li, Li, Zhao, Wu, Duan and Wang (2022); Zhao, Li, Dong and Zhao (2018).

Although image similarity-based methods are straightforward, they frequently encounter challenges when it comes to achieving precise reconstruction. Although image similarity-based methods are straightforward, they frequently encounter challenges when it comes to achieving precise reconstruction. To address that, the knowledge distillation technique Croitoru, Ristea, Dăscălescu, Ionescu, Khan and Shah (2024); Deng and Li (2022); Li, He, Li and Zhang (2023); Qin, Gu, Yu and Zhang (2023) was applied to reconstruct images. The disparities between the teacher and student networks can help to detect anomalies. Another notable approach is Normalizing Flow-based methods Gudovskiy, Ishizaka and Kozuka (2022); Yu, Zheng, Wang, Li, Wu, Zhao and Liwei (2021), employed an invertible transformation to transform image features into a normal distribution, where anomalous features tend to be presented as outliers. In addition, recent studies have shown that a simple memory bank approach Roth et al. (2022) can achieve remarkable performance in anomaly detection.

2.2. 3D Anomaly Detection

Numerous image anomaly detection methodologies have been effectively adapted for point cloud anomaly detection, including those based on knowledge distillation Masuda, Hachiuma, Fujii, Saito and Sekikawa (2021b); Qin et al. (2023) and reconstruction techniques Li, Xu, Gu, Zheng, Gao and Wu (2024); Zhou, Wang, Fang, Wang, Qiu and Zhang (2025). Notably, prototype-based methods Cao, Xu and Shen (2024); Chu, Chieh, Hsieh, Chen and Liu (2023); Horwitz and Hoshen (2022); Wang, Peng, Zhang, Yi, Wang and Wang (2023) have emerged as the predominant approach in this domain. These methods primarily distinguish themselves through their feature extraction strategies. For instance, BTF Horwitz and Hoshen (2022) utilizes handcrafted Fast Point Feature Histograms (FPFH), while M3DM Wang et al. (2023) employs features derived from a pre-trained PointMAE Pang, Wang, Tay, Liu, Tian and Yuan (2022b) model. To optimize feature extraction for specific datasets, Shape-guided Chu et al. (2023) and Group3AD Zhu, Xie, Hou, Dai, Gao, Wang and Shen (2024) enhance feature quality through self-supervised learning, fine-tuning the pre-trained point encoder. Furthermore, leveraging the robust capabilities of pre-trained image encoders, several approaches convert point clouds into multi-view images, subsequently applying these encoders Cao et al. (2024); Cheng, Cao, Xie, Lu and Shen (2024); Zhou et al. (2025) for point-wise feature extraction.

The M3DM model Wang et al. (2023) successfully combined point cloud data projection with image features for anomaly score computation. The model did not achieve satisfactory results on the Real3D-AD dataset because the projection operation caused a decrease in point cloud data density. In contrast, the Reg3D-AD model Liu et al. (2023) directly conducted feature selection and matching at the point level, and used the PatchCore architecture to achieve superior performance. While PatchCore architecture was proven effective in 2D anomaly detection, it may not be the best choice for detecting anomalies in 3D point clouds. Positional information of training and test image pixels is typically less important for 2D anomaly detection, as variations can occur due to different shooting angles. As a result, it is essential to create separate memory banks for various feature descriptors. Positional information is crucial for detecting anomalies in 3D, even though it may be challenging to spot in point cloud datasets. Choosing a suitable point cloud registration algorithm allows the training and test point clouds to be aligned in the same position. This encourages us to search for effective and reliable methods for detecting anomalies in 3D data to extract and utilize positional information.

3. Method

This paper introduces PointCore, a tool for identifying flaws in provided point cloud data, and our research focuses on investigating global-local registration and creating a memory bank. The pipeline is overviewed in Fig. 1.

3.1. Global and Local Registration

Similar to the Reg3D-AD model Liu et al. (2023), we apply the FPFH Rusu, Blodow and Beetz (2009) feature descriptor and the Random sample consensus (RANSAC) algorithm to achieve global registration (GR) of point clouds. To enhance the stability of point cloud registration, a point-plane ICP Rusinkiewicz and Levoy (2001) is introduced to locally optimize the outputs from global registration. Suppose two point clouds X_s (source point cloud) and X_t (target point cloud) need to be registered, the procedure of point-plane ICP is as follows:

1. Utilize the rotation matrix and translation vector from the global registration to convert X_s.
2. Search q_i in X_t such that closest to p_i in X_s, where the normal vector of q_i is denoted as n_i.
3. Assume optimal rotation Euler angles $\alpha, \beta, \gamma \to 0$, we have $\cos(\theta) \to 1, \sin(\theta) \to 0, \theta \to 0$. The rotation matrix R can be approximately expressed as:

$$R \approx \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} \tag{1}$$

4. Assume the optimal translation vector is $t = \left[t_x, t_y, t_z\right]$. The loss function can be shown as eq. 2. Compute the least squares problem via Moore–Penrose inverse.

$$E(R,t) = \sum_{i=1}^{n} \left(\left(Rp_i + t - q_i \right)^T n_i \right)^2 \tag{2}$$

5. Apply the computed rotation matrix and translation vector to transform X_s, and repeat steps 2-5 until the loss value is below a predefined threshold. Note that since the X_t is the target point cloud for registration, it is assumed to be fixed.

Table 1

Comparison of characteristics between PointCore and representative related works. Our method uniquely combines global-local registration with a unified coordinate-semantic memory bank.

Method	Modality	Core Approach	Registration	Key Characteristics
2D Image Anomaly Detection				
PaDiM (Defard et al., 2020)	2D	Memory Bank		Gaussian modeling of patch embeddings
FastFlow (Yu et al., 2021)	2D	Normalizing Flow		2D flow-based probability estimation
CFLOW-AD (Gudovskiy et al., 2022)	2D	Normalizing Flow		Conditional normalizing flows with positional encoding
PatchCore (Roth et al., 2022)	2D	Memory Bank		Coreset subsampling; Nearest neighbor search
3D Reconstruction & Projection Methods				
IMRNet (Li and Xu, 2023)	3D	Reconstruction		3D anomaly synthesis for self-supervised learning
FoldingNet-based VAE (Masuda et al., 2021a)	3D	Reconstruction		VAE-based point cloud reconstruction
CPMF (Cao et al., 2024)	3D (proj. 2D)	Projection		Projects 3D to 2D planes; Pseudo-multimodal features
R3D-AD (Zhou et al., 2025)	3D	Reconstruction		Reconstruction via diffusion models
3D Feature & Memory Bank Methods				
BTF (Horwitz and Hoshen, 2022)	3D	Handcrafted Feature		Uses classical FPFH features; No deep networks
M3DM (Wang et al., 2023)	3D + RGB	Hybrid Memory		Hybrid fusion of PointMAE and Image features; Split banks
Group3AD (Zhu et al., 2024)	3D	Feature Contrast		Group-level feature contrastive learning
3D-MMFN (Asad, Azeem, Malik, Jiang, Ali, Yang and Liu, 2025)	3D + RGB	Multimodal Fusion		Multi-level fusion of visual and geometric features
Registration-Based Methods				
Reg3D-AD (Liu et al., 2023)	3D	Memory Bank	(Global)	**Split** memory banks (Local/Global); Susceptible to mismatch
PointCore (Ours)	3D	Memory Bank	(Global+Local)	**Unified** Coordinate-Semantic bank; Hierarchical search

3.2. Memory Bank Construction

Coordinate Sampling. A reasonable method for downsampling point clouds is necessary to accelerate the model's inference speed due to the large amount of coordinate information in point cloud data. In this paper, we employ a greedy down-sampling algorithm Yang and Guo (2016) to sample the point clouds. Given the point clouds X, and the point set C_a, a is the number of points in C_a. We aim to obtain $S_{\max}$ uniform points from C_a. The specific procedure is as follows:

1. Randomly select S_{init} points from C_a to construct the initial point set $P_{\text{init}} = \left\{ P_1, P_2, P_3, \dots, P_{S_{\text{init}}} \right\}$.
2. Calculate distances between C_a and P_{init} to obtain a matrix d_{2d} with dimensions $a \times S_{\text{init}}$.

$$d_{2d} = \begin{bmatrix} d_{11} & \cdots & d_{1S_{\text{init}}} \\ \vdots & \ddots & \vdots \\ d_{a1} & \cdots & d_{aS_{\text{init}}} \end{bmatrix}$$

3. Calculate the mean for each row of the matrix d_{2d} to obtain d_{1d}.

$$d_{1d} = [d_{1_{\text{mean}}}, d_{2_{\text{mean}}} \cdots, d_{a_{\text{mean}}}]$$

4. Locate the maximum value in the matrix d_{1d}, and add the corresponding point to P_{init}. Repeat steps 2-4 until the number of elements in P_{init} equals $S_{\max}$.

Point Feature Interpolation. First, a point transformer (PointMAE) Pang et al. (2022b); Zhao, Jiang, Jia, Torr and Koltun (2021) pretrained on the ShapeNet Chang, Funkhouser, Guibas, Hanrahan, Huang, Li, Savarese, Savva, Song, Su, Xiao, Yi and Yu (2015a) dataset is utilized as our 3D feature extractor. For a fair comparison, the outputs of layers $\{3, 7, 11\}$ are likewise selected as our 3D feature. The point transformer encodes point clouds into point groups by assigning each group a center point for position and neighbor numbers for group size. We utilize the coordinates collected during the coordinate sampling stage as the center points for groups in each point cloud. Every key point serves as a memory element that is connected to both coordinates and PointMAE characteristics. Throughout the model training process, we extract tens of thousands of memory elements from every template point cloud. These components make up the memory storage in the pipeline, as shown in Fig. 1. Downsampling the point cloud coordinates is necessary to decrease the computational complexity of PointMAE feature computation during inference. As a result, we conduct point feature interpolation to assign a PointMAE feature value to every coordinate in the feature bank. This process of the feature interpolation method is shown in Fig. 2.

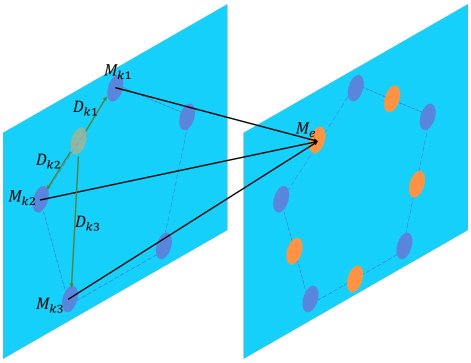
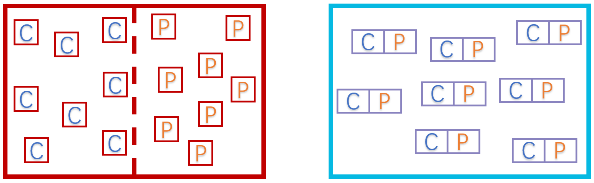


Figure 2: The process of the PointMAE feature interpolation.

The set P_{central} contains the coordinates of all central points, and the corresponding feature set is M_{init}. Taking a non-central point P_e as an example, we use the k-nearest neighbors algorithm to obtain three nearest neighbors $\left[P_{k1}, P_{k2}, P_{k3}\right]$ in the point set P_{central}. The corresponding Euclidean distances and PointMAE feature values are



(a) PatchCore (b) PointCore(ours)

Figure 3: The differences between PointCore (ours) and PatchCore memory bank architectures. (C in the figure represents the coordinates, and P represents the PointMAE feature.)

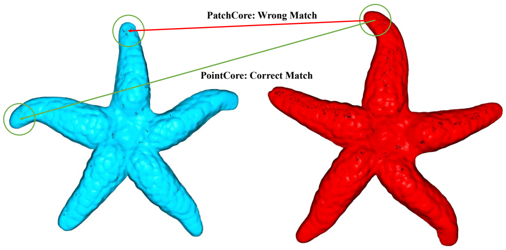


Figure 4: Errors in matching caused by similar local features.

$\left[D_{k1}, D_{k2}, D_{k3}\right]$ and $\left[M_{k1}, M_{k2}, M_{k3}\right]$, respectively. By using eq. (3), we obtain the PointMAE feature M_e for P_e. This process is repeated until all non-central points obtain their corresponding PointMAE features. The M_e is computed by

$$M_e = \frac{D_{k1}D_{k2}M_{k3} + D_{k1}D_{k3}M_{k2} + D_{k2}D_{k3}M_{k1}}{D_{k1}D_{k2} + D_{k1}D_{k3} + D_{k2}D_{k3}} \tag{3}$$

Memory Bank Architecture. The memory bank architecture, as shown in Figure 3, integrates coordinate information and feature values from point clouds. This helps decrease the computational expense of nearest-neighbor calculations for features and also avoids substantial errors in feature matching due to local similarities, as shown in Fig. 4.

Rationale for the Unified Coordinate-Semantic Memory Bank. Our decision to combine low-dimensional coordinate features (3D) and high-dimensional semantic features (1154D) into a single, unified memory bank is a deliberate design choice aimed at overcoming the fundamental limitations of previous methods. The core justification rests on a hierarchical matching strategy that leverages spatial proximity as a powerful prior for semantic feature retrieval. Since deep local features can be ambiguous across different parts of a rigid object (e.g., similar flat surfaces at different locations), coordinate constraints effectively resolve this ambiguity by enforcing structural consistency. Instead of performing an expensive and often misleading nearest-neighbor search directly in the high-dimensional feature space, our approach first uses the 3D coordinates to efficiently identify a small subset of k spatially closest candidate points. This initial step is computationally inexpensive and effectively grounds the search in the geometric reality of the point cloud. Only within this geometrically constrained candidate set do we then compute distances in the 1154D PointMAE feature space to find the single best semantic match. This two-stage process directly addresses the "feature mismatch" problem illustrated in Fig. 4, where methods like PatchCore, which use separate memory banks, can erroneously pair semantically similar but spatially distant features, violating the object's structural integrity. Our unified architecture, therefore, is not a simple concatenation of features but a principled mechanism tailored for point cloud data, where spatial coherence is not just a feature but a fundamental organizing principle.

3.3. Inference Block

Multi-feature Anomaly Score Calculation. To address the concern about using a consistent metric across different feature dimensions, we explicitly calculate anomaly scores in the coordinate and semantic spaces separately before fusing them. This avoids the direct, and potentially misleading, combination of distances from disparate feature spaces. Unlike multi-modal fusion methods that require explicit alignment of heterogeneous features into a shared latent space, our framework relies on the inherent structural correspondence of the point cloud. Specifically, each coordinate vector in the memory bank is physically tied to its co-located semantic feature vector. Thus, no learnable alignment or projection is needed; the matching is performed hierarchically by using coordinates to index the relevant semantic features. The memory bank is composed of the element sets, that is,

$$M_{\text{train}} = \{(M_{1_c}, M_{1_p}), (M_{2_c}, M_{2_p}), \ldots, (M_{n_c}, M_{n_p})\} \tag{4}$$

where M_{i_c} denotes the coordinates of the i-th point, and M_{i_p} represents the PointMAE feature of the i-th point. The test feature bank is defined as

$$F_{\text{test}} = \{(F_{1_c}, F_{1_p}), (F_{2_c}, F_{2_p}), \ldots, (F_{m_c}, F_{m_p})\} \tag{5}$$

where F_{j_c} is the coordinates of the j-th point, and F_{j_p} is the PointMAE feature of the j-th point.

For an element $\left(F_{j_c}, F_{j_p}\right)$ in F_{test}, we use its coordinate information F_{j_c} to find three nearest neighbors in M_{train}, denoted as $\{\left(M_{i_c}, M_{i_p}\right), \left(M_{o_c}, M_{o_p}\right), \left(M_{u_c}, M_{u_p}\right)\}$. Using Euclidean distance, their coordinate distances $\{\mathrm{DC}_1, \mathrm{DC}_2, \mathrm{DC}_3\}$ and feature distances $\{\mathrm{DP}_1, \mathrm{DP}_2, \mathrm{DP}_3\}$ are obtained. The final coordinate anomaly score S_c and PointMAE anomaly score S_p are calculated by

$$\begin{aligned} S_c &= \mathrm{mean}(\mathrm{DC}_1, \mathrm{DC}_2, \mathrm{DC}_3), \\ S_p &= \min(\mathrm{DP}_1, \mathrm{DP}_2, \mathrm{DP}_3). \end{aligned} \tag{6}$$

Ranking Block. Both anomaly scores are required for normalization due to variations in scale and distribution. The interval scaling method is commonly used in traditional normalization techniques. For a set of data S_{list}, the interval scaling process is shown

$$S_{\text{norm}} = \frac{S_{\text{list}} - \min(S_{\text{list}})}{\max(S_{\text{list}}) - \min(S_{\text{list}})} \tag{7}$$

Table 2
O-AUROC score (↑) on Real3D-AD. The best and second-best results are marked in red and blue, respectively.

Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Average
BTF(Raw)	0.520	0.560	0.462	0.432	0.545	0.784	0.549	0.648	0.779	0.754	0.575	0.630	0.603
BTF(FPFH)	0.730	0.647	0.703	0.789	0.707	0.691	0.602	0.686	0.596	0.396	0.530	0.539	0.635
M3DM(PointBERT)	0.407	0.506	0.442	0.673	0.627	0.466	0.556	0.617	0.494	0.577	0.528	0.562	0.538
M3DM(PointMAE)	0.434	0.541	0.450	0.683	0.602	0.433	0.540	0.644	0.495	0.694	0.551	0.552	0.552
PatchCore(FPFH)	0.882	0.590	0.565	0.837	0.574	0.546	0.675	0.370	0.505	0.589	0.441	0.541	0.593
PatchCore(FPFH+Raw)	0.848	0.777	0.626	0.853	0.784	0.628	0.837	0.359	0.767	0.663	0.471	0.570	0.682
PatchCore(PointMAE)	0.726	0.498	0.585	0.827	0.783	0.489	0.630	0.374	0.539	0.501	0.519	0.663	0.594
CPMF	0.632	0.518	0.718	0.640	0.640	0.554	0.840	0.349	0.843	0.393	0.526	0.845	0.625
IMRNet	0.762	0.711	0.755	0.780	0.905	0.517	0.880	0.674	0.604	0.665	0.674	0.774	0.725
Reg3D-AD	0.716	0.697	0.827	0.852	0.900	0.584	0.915	0.417	0.762	0.583	0.506	0.685	0.704
Group3AD	0.744	0.728	0.847	0.786	0.932	0.679	0.976	0.539	0.841	0.585	0.562	0.796	0.751
PointCore(FPFH+Raw)	0.792	0.871	0.861	0.842	0.847	0.642	0.915	0.477	0.954	0.853	0.617	0.728	0.783
PointCore(PointMAE+Raw)	0.660	0.866	0.976	0.841	0.963	0.684	0.993	0.535	0.973	0.882	0.652	0.929	0.829

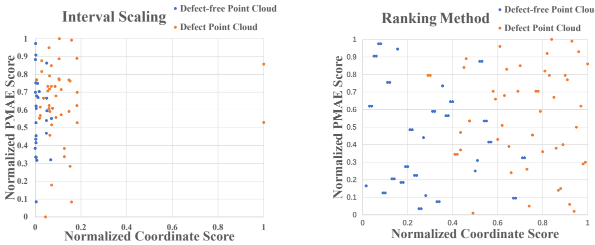
Table 3
O-AUPR score (↑) on Real3D-AD. The best and second-best results are marked in red and blue, respectively.

Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Average
BTF(Raw)	0.506	0.523	0.490	0.464	0.535	0.760	0.633	0.598	0.793	0.751	0.579	0.700	0.611
BTF(FPFH)	0.659	0.653	0.638	0.814	0.677	0.620	0.638	0.603	0.567	0.434	0.557	0.505	0.614
M3DM(PointBERT)	0.497	0.517	0.480	0.716	0.661	0.569	0.628	0.628	0.491	0.638	0.573	0.569	0.581
M3DM(PointMAE)	0.479	0.508	0.498	0.739	0.620	0.533	0.525	0.663	0.518	0.616	0.573	0.593	0.572
PatchCore(FPFH)	0.852	0.611	0.553	0.872	0.569	0.506	0.642	0.411	0.508	0.573	0.491	0.506	0.591
PatchCore(FPFH+Raw)	0.807	0.766	0.611	0.885	0.767	0.560	0.844	0.411	0.763	0.553	0.473	0.559	0.667
PatchCore(PointMAE)	0.747	0.555	0.576	0.864	0.801	0.488	0.720	0.444	0.546	0.590	0.561	0.708	0.633
Reg3D-AD	0.703	0.753	0.824	0.884	0.884	0.588	0.939	0.454	0.787	0.646	0.491	0.721	0.723
Group3AD	0.757	0.706	0.837	0.674	0.932	0.612	0.981	0.533	0.842	0.648	0.567	0.785	0.740
PointCore	0.667	0.862	0.973	0.863	0.957	0.623	0.993	0.548	0.972	0.774	0.585	0.938	0.813

As shown in Fig. 5, the interval scaling technique can standardize variations in scale between two anomaly scores but is unable to handle variations in distribution. If there are two outliers in the coordinate anomaly score, the resulting anomaly score will be significantly lower compared to the PointMAE anomaly score. This greatly affects ensemble strategies that rely on mathematical operations.

To address this issue, we have designed a ranking-based normalization method, as outlined in eq.(8), where $Sort\_rank(S_{\text{list}})$ obtains the ranking of each value in S_{list}, and $len(S_{\text{list}})$ is the length of S_{list}.

$$S_{\text{norm}} = \frac{Sort\_rank(S_{\text{list}})}{len(S_{\text{list}})} \tag{8}$$



(a) Traditional normalization method.

(b) Ranking-based normalization method.

Figure 5: Distribution of coordinate and PointMAE anomaly scores normalized by these two methods.

Table 4
P-AUROC score (↑) on Real3D-AD. The best and second-best results are marked in red and blue, respectively.

Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Average
BTF(Raw)	0.564	0.647	0.735	0.608	0.563	0.601	0.514	0.597	0.520	0.489	0.392	0.623	0.571
BTF(FPFH)	0.738	0.708	0.864	0.693	0.882	0.875	0.709	0.891	0.512	0.571	0.501	0.815	0.730
M3DM(PointBERT)	0.523	0.593	0.682	0.790	0.594	0.668	0.589	0.646	0.574	0.732	0.563	0.677	0.636
M3DM(PointMAE)	0.530	0.607	0.683	0.735	0.618	0.678	0.600	0.654	0.561	0.748	0.555	0.679	0.637
PatchCore(FPFH)	0.471	0.643	0.637	0.618	0.760	0.430	0.464	0.830	0.544	0.596	0.522	0.411	0.577
PatchCore(FPFH+Raw)	0.556	0.740	0.749	0.558	0.854	0.658	0.781	0.539	0.808	0.753	0.613	0.549	0.680
PatchCore(PointMAE)	0.579	0.610	0.635	0.683	0.776	0.439	0.714	0.514	0.660	0.725	0.641	0.727	0.642
CPMF	0.618	0.836	0.734	0.559	0.753	0.719	0.988	0.449	0.962	0.725	0.800	0.959	0.758
Reg3D-AD	0.631	0.718	0.724	0.676	0.835	0.503	0.826	0.545	0.817	0.811	0.617	0.759	0.705
Group3AD	0.636	0.745	0.738	0.759	0.862	0.631	0.836	0.564	0.827	0.798	0.625	0.803	0.735
PointCore	0.608	0.706	0.706	0.780	0.810	0.712	0.782	0.515	0.841	0.781	0.736	0.745	0.731

Table 5
P-AUPR score (↑) on Real3D-AD. The best and second-best results are marked in red and blue, respectively.

Method	Airplane	Car	Candybar	Chicken	Diamond	Duck	Fish	Gemstone	Seahorse	Shell	Starfish	Toffees	Average
BTF(Raw)	0.012	0.014	0.025	0.049	0.032	0.020	0.017	0.014	0.031	0.011	0.017	0.016	0.022
BTF(FPFH)	0.027	0.028	0.118	0.044	0.239	0.068	0.036	0.075	0.027	0.018	0.034	0.055	0.064
M3DM(PointBERT)	0.007	0.017	0.016	0.377	0.038	0.011	0.039	0.017	0.028	0.021	0.040	0.018	0.052
M3DM(PointMAE)	0.007	0.018	0.016	0.310	0.033	0.011	0.025	0.018	0.030	0.022	0.040	0.021	0.046
PatchCore(FPFH)	0.027	0.034	0.142	0.040	0.273	0.055	0.052	0.093	0.031	0.031	0.037	0.040	0.071
PatchCore(FPFH+Raw)	0.016	0.160	0.092	0.045	0.363	0.034	0.266	0.066	0.291	0.049	0.035	0.055	0.123
PatchCore(PointMAE)	0.016	0.069	0.020	0.052	0.107	0.008	0.201	0.008	0.071	0.043	0.046	0.055	0.058
CPMF	0.010	0.064	0.050	0.031	0.074	0.018	0.559	0.007	0.636	0.025	0.128	0.391	0.166
Reg3D-AD	0.017	0.135	0.109	0.044	0.191	0.010	0.437	0.016	0.182	0.065	0.039	0.067	0.109
Group3AD	0.018	0.174	0.122	0.068	0.287	0.016	0.448	0.009	0.24	0.067	0.056	0.134	0.137
PointCore	0.016	0.088	0.322	0.413	0.493	0.044	0.510	0.007	0.637	0.086	0.048	0.347	0.251

4. Experiments

4.1. Experimental Details

Dataset. We conduct extensive experiments on two datasets: **Real3D-AD** Liu et al. (2023), **Anomaly-ShapeNet** Li et al. (2024), and the recently proposed **MulSen-AD ?**. The Real3D-AD Liu et al. (2023) dataset contains 1,254 samples divided into 12 different categories. In each training set for a particular category, there are only four samples, resembling the few-shot situation in 2D anomaly detection. These categories include Airplane, Candybar, Chicken, Diamond, Duck, Fish, Gemstone, Seahorse, Shell, Starfish, and Toffees. All of these categories include toys made from production lines. Its noteworthy achievement includes a point resolution of 0.04mm and a precision of 0.011mm. This is notably higher than other previous point cloud anomaly datasets. Moreover, the system's effectiveness is enhanced by incorporating multi-view scanning, which helps to eliminate blind spots and improve its ability to detect anomalies. As a result, Real3D-AD is considered more appropriate for achieving accurate point cloud anomaly detection.

Anomaly-ShapeNet provides diverse synthetic anomalies generated on ShapeNet Chang, Funkhouser, Guibas, Hanrahan, Huang, Li, Savarese, Savva, Song, Su et al. (2015b), which offer 40 object classes, respectively. Each object class includes normal training samples and test samples containing both normal and abnormal instances. For each test sample, detailed ground truth annotations are provided, including point-wise masks and object-wise labels.

MulSen-AD ? is a multi-sensor industrial anomaly detection dataset that includes high-resolution RGB images, infrared thermal images, and high-precision 3D point cloud data. It spans 15 diverse industrial product categories with a wide range of real-world anomalies. To maintain a fair comparison with other point cloud-based methods, we evaluate our framework using only the point cloud modality from this dataset.

Our primary experiments focus on Real3D-AD, with Anomaly-ShapeNet and MulSen-AD results included for additional insights.

Baselines. We compare with BTF Horwitz and Hoshen (2022), M3DM Wang et al. (2023), PatchCore Roth et al. (2022), Reg3D-AD Liu et al. (2023), CPMF Cao et al. (2024), IMRNet Li et al. (2024), Group3AD Zhu et al. (2024) to the remarkable performance of our method. According to the used point cloud features, they can be grouped into 11 different baselines, namely, BTF (Raw), BTF (FPFH), M3DM (PointMAE), PatchCore (FPFH), PatchCore (FPFH+ Raw), PatchCore (PointMAE), CPMF, IMRNet, Reg3D-AD, Group3AD, where Raw denotes using coordinate information, PointMAE and FPFH are two different feature descriptors.

Evaluation Metrics. All evaluation metrics are the same as in Liu et al. (2023). We evaluate the object-level anomaly detection performance and the point-level anomaly detection performance via the area under the receiver operator curve (AUROC) and the area under the Precision-Recall curve (AUPR/AP). The higher the AUROC and AUPR, the better anomaly detection performance is.

Implementation Details. For the sake of reproducibility and to ensure a fair comparison with baseline methods, we detail our key implementation parameters in Table 6. Where applicable, parameters related to feature extraction and memory bank construction are intentionally aligned with those used in Reg3D-AD Liu et al. (2023) to isolate the performance gains from our architectural contributions. It is important to clarify that our framework is fully unsupervised, meaning no anomaly labels were used during the training phase. Consequently, there is no "training accuracy" to optimize. To rigorously avoid overfitting and data leakage, all key hyperparameters (e.g., coreset percentage 1%) were adopted directly from the baseline method Reg3D-AD without further tuning on the test set. Our novel components, such as the point-to-plane ICP local optimization, are configured with standard, widely-accepted values to demonstrate their general effectiveness. A fixed random seed was used across all experiments to ensure consistent results from stochastic processes like point sampling and RANSAC.

Experimental Environment. All experiments are conducted on 12th Gen Intel(R) Core(TM) i9-12900K CPU, 64G DDR4 SDRAM and GeForce RTX 3090 platform.

Table 6
Key hyperparameters and implementation details for our PointCore framework. Settings are chosen to ensure reproducibility and fair comparison with prior work.

Component	Parameter	Value	Justification
Registration	Global Reg. (RANSAC) - Distance Threshold - Max Iterations - Confidence	 0.05 4,000,000 0.999	Set based on point cloud resolution and a high confidence requirement to ensure robust initial alignment even with numerous outlier feature matches.
	Local Opt. (ICP) - Max Iterations - Convergence Tol.	 30 1e-6	Standard values in point cloud registration that balance precision and efficiency, ensuring high-quality alignment.
Feature Extraction	Sampled Points ($S_{\max}$)	10,000	Balances representative surface coverage with computational tractability for subsequent feature processing.
	PointMAE Layers	{3, 7, 11}	To maintain a **fair comparison** with Reg3D-AD, which uses the same multi-level feature configuration.
Memory & Inference	Nearest Neighbors (k)	3	Provides a stable local neighborhood for both feature interpolation and anomaly score matching while minimizing computational overhead.
	Coreset Percentage	1%	Aligned with Reg3D-AD and PatchCore to ensure a **fair comparison** of both performance and inference speed.
Reproducibility	Random Seed	42	Ensures the **reproducibility** of our experimental results across all stochastic processes.

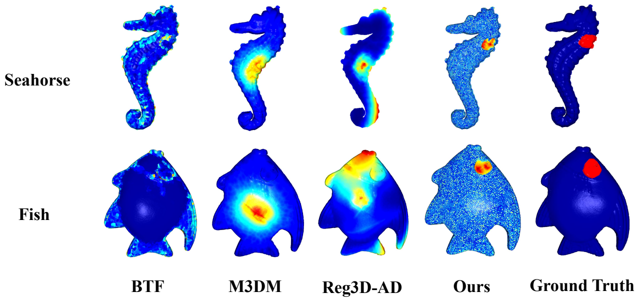


Figure 6: Heatmaps of anomaly scores obtained by several methods on the Real3D-AD dataset. The visualization demonstrates that the proposed method is more accurate in detecting and locating anomaly data points than other methods.

4.2. Anomaly Detection on Real3D-AD and Anomaly-ShapeNet Datasets

We compare our method with several methods on Real3D-AD, and Table 2 shows the anomaly detection results with O-AUROC(object-level AUROC). The FPFH Rusu, Blodow, Marton and Beetz (2008) and PointMAE features are combined with the Raw feature, respectively. The findings suggest that the PointMAE-based combination outperforms others. The coordinate information of the point cloud is essential for the PointCore architecture being proposed. Tables 2 to 5 presents a more comprehensive comparison. PointCore has achieved competitive performance across all metrics and has obtained a 10.38% improvement in the O-AUROC metric. The performance in point-level AUROC (P-AUROC), object-level AUPR (O-AUPR), and point-level AUPR (P-AUPR) further demonstrate the superior performance of our proposal in anomaly detection. (Memory bank subsampling for all methods is configured at 1%.). We visualize some representative sample of Real3D-AD for anomaly detection and localization in Figure 6.

Anomaly detection results on Anomaly-ShapeNet are shown in Table 7. Compared to other methods, our PointCore performs better on the average O-AUROC, which achieves 67.7% on Anomaly-ShapeNet dataset. Our PointCore achieves the highest score for 17 out of 40 classes.

4.3. Results on the MulSen-AD Dataset

To further verify the generalization ability and robustness of our proposed PointCore framework, we conducted experiments on the newly released MulSen-AD dataset **?**. We follow the original paper's evaluation protocol and report the object-level AUROC (O-AUROC) using only the point cloud data. The results, compared with several state-of-the-art point cloud-based anomaly detection methods, are presented in Table 9. As shown in the table, PointCore achieves a competitive mean O-AUROC score, demonstrating its effectiveness on this challenging dataset. Notably, our method outperforms other strong baselines such as Reg3D-AD and M3DM, and achieves the best performance in several categories like Piggy and Button Cell. This indicates that our unified local-global feature representation and robust score normalization strategy can be well generalized to different types of industrial products and anomalies.

4.4. Inference Time

The comparisons of inference time and performance among BTF, M3DM, and PatchCore are implemented, as tabulated in Table 8. These results were obtained by averaging over multiple runs with different random seeds to ensure robustness. As can be seen, BTF presented in Table 8, albeit

Table 7
O-AUROC score for anomaly detection of 40 categories of Anomaly-ShapeNet dataset. The best and second-best results are marked in red and blue, respectively.

Method	cap0	cap3	helmet3	cup0	bowl4	vase3	headset1	eraser0	vase8	cap4	vase2	vase4	helmet0	bucket1
BTF(Raw)	0.668	0.527	0.526	0.403	0.664	0.717	0.515	0.525	0.424	0.468	0.410	0.425	0.553	0.321
BTF(FPFH)	0.618	0.522	0.444	0.586	0.609	0.699	0.490	0.719	0.668	0.520	0.546	0.510	0.571	0.633
M3DM	0.557	0.423	0.374	0.539	0.464	0.439	0.617	0.627	0.663	0.777	0.737	0.476	0.526	0.501
Patchcore(FPFH)	0.580	0.453	0.404	0.600	0.494	0.449	0.637	0.657	0.662	0.757	0.721	0.506	0.546	0.551
Patchcore(PointMAE)	0.589	0.476	0.424	0.610	0.501	0.460	0.627	0.677	0.663	0.727	0.741	0.516	0.556	0.561
CPMF	0.601	0.551	0.520	0.497	0.683	0.582	0.458	0.689	0.529	0.553	0.582	0.514	0.555	0.601
Reg3D-AD	0.693	0.725	0.367	0.510	0.663	0.650	0.610	0.343	0.620	0.643	0.605	0.500	0.600	0.752
IMRNet	0.737	0.775	0.573	0.643	0.676	0.700	0.676	0.548	0.630	0.652	0.614	0.524	0.597	0.771
PointCore	0.511	0.669	0.536	0.550	0.741	0.698	0.876	0.750	0.717	0.514	0.920	0.624	0.477	0.711

Method	bottle3	vase0	bottle0	tap1	bowl0	bucket0	vase5	vase1	vase9	ashtray0	bottle1	tap0	phone	cup1
BTF(Raw)	0.568	0.531	0.597	0.573	0.564	0.617	0.585	0.549	0.564	0.578	0.510	0.525	0.563	0.521
BTF(FPFH)	0.322	0.342	0.344	0.546	0.509	0.401	0.409	0.219	0.268	0.420	0.546	0.560	0.671	0.610
M3DM	0.541	0.423	0.574	0.739	0.634	0.309	0.317	0.427	0.663	0.577	0.637	0.754	0.357	0.556
Patchcore(FPFH)	0.572	0.455	0.604	0.766	0.504	0.469	0.417	0.423	0.660	0.587	0.667	0.753	0.388	0.586
Patchcore(PointMAE)	0.650	0.447	0.513	0.538	0.523	0.593	0.579	0.552	0.629	0.591	0.601	0.458	0.488	0.556
CPMF	0.405	0.451	0.520	0.697	0.784	0.482	0.618	0.345	0.609	0.353	0.482	0.359	0.509	0.499
Reg3D-AD	0.525	0.533	0.486	0.641	0.671	0.610	0.520	0.702	0.594	0.597	0.695	0.676	0.414	0.538
IMRNet	0.640	0.533	0.552	0.696	0.681	0.580	0.676	0.757	0.594	0.671	0.700	0.676	0.755	0.757
PointCore	0.635	0.525	0.709	0.800	0.830	0.575	0.710	0.612	0.800	1.000	1.000	0.645	0.662	0.563

Method	vase7	helmet2	cap5	shelf0	bowl5	bowl3	helmet1	bowl1	headset0	bag0	bowl2	jar	**Mean**
BTF(Raw)	0.448	0.602	0.373	0.164	0.417	0.385	0.349	0.264	0.378	0.410	0.525	0.420	0.493
BTF(FPFH)	0.518	0.542	0.586	0.609	0.699	0.490	0.719	0.668	0.520	0.546	0.510	0.424	0.528
M3DM	0.657	0.623	0.639	0.564	0.409	0.617	0.427	0.663	0.577	0.537	0.684	0.441	0.552
Patchcore(FPFH)	0.693	0.425	0.790	0.494	0.558	0.537	0.484	0.639	0.583	0.571	0.615	0.472	0.568
Patchcore(PointMAE)	0.650	0.447	0.538	0.523	0.593	0.579	0.552	0.629	0.591	0.601	0.458	0.483	0.562
CPMF	0.397	0.462	0.697	0.685	0.685	0.658	0.589	0.639	0.643	0.643	0.625	0.610	0.559
Reg3D-AD	0.462	0.614	0.467	0.688	0.593	0.348	0.381	0.525	0.537	0.706	0.490	0.592	0.572
IMRNet	0.635	0.641	0.652	0.603	0.710	0.599	0.600	0.702	0.720	0.660	0.685	0.780	0.661
PointCore	0.567	0.527	0.689	0.701	0.553	0.744	0.414	0.774	0.711	0.557	0.570	0.900	0.677

Method	PointCore	BTF	M3DM	Reg3D-AD
O-AUROC	0.829 ± 0.03	0.603 ± 0.35	0.552 ± 0.41	0.704 ± 0.29
P-AUROC	0.731 ± 0.03	0.571 ± 0.38	0.637 ± 0.31	0.705 ± 0.25
Time(s)	4.282	3.882	5.061	13.022

Table 8
Mean inference time per object on Real3D-AD. The values are presented as mean ± standard deviation, obtained from multiple runs with different seeds. PointCore exhibits the highest stability across all metrics.

fast, shows mediocre to poor performance in O-AUROC and P-AUROC. The proposed PointCore achieves the competitive inference speed and the best scores. Notably, PointCore also demonstrates significantly higher stability, with a much smaller standard deviation across all metrics compared to the other methods, which exhibit larger fluctuations.

4.5. Ablation Studies

In this section, we ablate our design choices on the Reg3D-AD dataset under the 3-view settings, namely, LR (Local Optimization), PointCore, and RB (Ranking Block).

Effectiveness of Local Optimization. The Reg3D-AD model uses the FPFH+RANSAC method Rusu et al. (2009) for registration, with the RANSAC algorithm iteratively determining the best pose matrix. Nevertheless, the unpredictable selection of points by RANSAC in each iteration results in considerable variation in the final registration outcomes. As shown in Table 11, we perform 20 experiments using the registration algorithm on a pair of point clouds from the Toffees dataset. The outcomes show significant differences in the results produced by the RANSAC algorithm, despite having the same parameters.

The stability of the registration process greatly improved after the introduction of the local optimization algorithm (point-plane ICP). The variance in Euler angles decrease from 1.1058° to $(7.7796 \times 10^{-6})°$, allowing us to achieve a score of 0.642 ± 0.01. Although we were able to create a more stable model, its performance was slightly lower. It should be noted that the decrease in metrics is not because of the local optimization approach, but rather a fundamental problem in the Reg3D-AD architecture. By making further adjustments, we can improve the model's stability, speed, and overall performance.

We test the registration error of Point-Point ICP and Point-Plane ICP algorithms using different levels of Gaussian noise as part of our evaluation of the local optimization method. The results can be seen in Figure 7. The point-plane ICP registration method typically produced more precise results when the noise levels were reduced. The anomaly samples in the Reg3D-AD dataset resemble a point cloud with low noise levels. Therefore, the point-to-plane ICP algorithm is selected for local optimization. From the Table 10, it is evident that this choice is indeed correct.

Effectiveness of PointCore Architecture. PointCore architecture outperforms the Reg3D-AD architecture by effectively utilizing coordinate information from point clouds

Table 9

Object-level AUROC (↑) comparison on the point cloud modality of the MulSen-AD dataset. The best and second-best results are marked in red and blue, respectively. Note that our PointCore results are based on the PointMAE+Raw configuration.

Method	Capsule	Cotton	Cube	Spring Pad	Screw	Screen	Piggy	Nut	Flat Pad	Plastic Cyl.	Zipper	Button Cell	Toothbrush	Solar Panel	Light	Mean
BTF(FPFH)	0.923	0.320	0.634	0.512	0.592	0.788	0.377	0.531	0.660	0.579	0.637	0.655	0.659	0.423	0.378	0.578
BTF(Raw)	0.829	0.775	0.447	0.383	0.908	0.584	0.360	0.459	0.373	0.404	0.479	0.645	0.924	0.308	0.442	0.555
M3DM(PointMAE)	0.835	0.435	0.615	0.808	0.629	0.494	0.667	0.590	0.797	0.675	0.744	0.697	0.803	0.695	0.756	0.683
M3DM(PointBERT)	0.604	0.548	0.192	0.308	0.787	0.953	0.167	0.586	0.910	0.264	0.649	0.571	0.890	0.541	0.428	0.560
PatchCore(FPFH)	0.898	0.228	0.759	0.763	0.742	0.900	0.830	0.714	0.850	0.771	0.917	0.884	0.917	0.818	0.764	0.784
PatchCore(FPFH+raw)	0.892	0.395	0.664	0.754	0.739	0.922	0.760	0.831	0.743	0.843	0.872	0.710	0.932	0.759	0.683	0.767
PatchCore(PointMAE)	0.835	0.435	0.615	0.808	0.629	0.494	0.667	0.590	0.797	0.675	0.744	0.697	0.803	0.695	0.756	0.683
Reg3D-AD	0.867	0.592	0.451	0.804	0.716	0.444	0.610	0.683	0.700	0.725	0.810	0.613	0.898	0.695	0.789	0.693
PointCore(Ours)	0.835	0.720	0.752	0.812	0.815	0.913	0.867	0.812	0.911	0.804	0.910	0.931	0.915	0.810	0.745	0.837

Table 10

The results obtained from ablation experiments on each module. Baseline represents Reg3D-AD, LR(P-P) represents local optimization based on the point-point ICP algorithm, LR represents local optimization based on the point-plane ICP algorithm, and RB represents Ranking Block.

Method	O-AUROC↑	P-AUROC↑	O-AUPR↑	P-AUPR↑	Time(s)↓
Baseline	0.704	0.705	0.723	0.109	13.022
Baseline+LR	0.642	0.635	0.661	0.098	13.531
Baseline+RB	0.734	0.706	0.733	0.127	13.057
Baseline+LR+RB	0.653	0.641	0.679	0.107	13.612
PointCore	0.735	0.716	0.739	0.138	**3.929**
PointCore+LR	0.796	**0.732**	0.782	0.226	4.256
PointCore+RB	0.785	0.725	0.763	0.136	3.945
PointCore+LR+RB	**0.829**	0.731	**0.813**	**0.251**	4.282
PointCore+LR(P-P)+RB	0.746	0.618	0.733	0.056	8.314

Table 11

The variance of Euler angles (the unit is degree) and translation vectors (the multiples of resolution, dimensionless) in multiple sets of registration results (GR: Global Registration, LR: Local Optimization).

Method	Rotation(°)	Translation
GR	1.1058	134.1
GR+LR	7.7796×10^{-6}	1.1227×10^{-3}

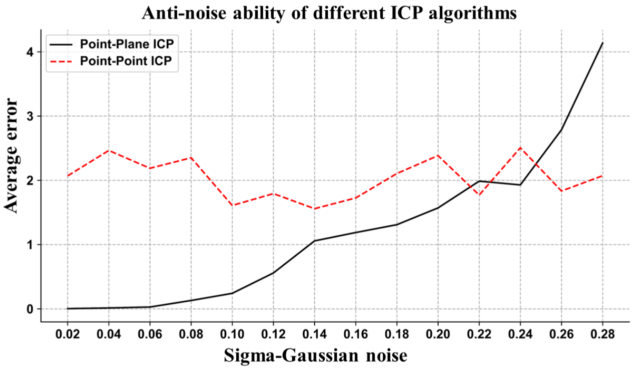


Figure 7: Tests are carried out on two registration algorithms with varying degrees of Gaussian noise. Each experiment is repeated 10 times.

to improve the speed and accuracy of anomaly detection. In terms of speed, we speed up the inference process by integrating the coordinate information and PointMAE feature information of points. This helps lower the high computational expense caused by the following PointMAE search for the closest neighbors. On the other hand, the Reg3D-AD design separates the coordinate information and PointMAE feature information into distinct memory banks. During inference, each coordinate and PointMAE must locate their closest neighbors in the memory bank. Dealing with the 1154 dimensions of the PointMAE feature presents a notable computational obstacle.

To improve accuracy, we prioritize coordinating information to prevent clear discrepancies in PointMAE characteristics. In the Reg3D-AD architecture, the PointMAE features of the test point cloud need to identify the closest neighbors in the PointMAE memory bank without using coordinate information. This process results in incorrect matches between groups that are similar locally, as depicted in Fig. 4. By narrowing down the range of coordinates that are considered a match, we greatly decrease the chances of errors.

Effectiveness of Combining Low-Dimensional and High-Dimensional Features. To quantitatively demonstrate the superiority of our design, we conducted a series of key ablation experiments, with the results presented in Table 12. We compared several variants: (1) **PointCore (Coordinates Only)**: using only the coordinate distance S_c as the anomaly score; (2) **PointCore (PointMAE Only)**: using only the

Table 12
Ablation study on different components of PointCore.

Method	O-AUROC ↑
PointCore (Coordinates Only)	0.785
PointCore (PointMAE Only)	0.721
PointCore (Naive Concatenation)	0.750
PointCore (Full)	**0.829**

PointMAE feature distance S_p as the anomaly score; (3) **PointCore (Naive Concatenation)**: performing a k-NN search after simply concatenating the 3D coordinates and 1154D features into a 1157D vector; and (4) **PointCore (Full)**. The results clearly show that our full model significantly outperforms all variants. Specifically, using only a single feature type (either coordinates or PointMAE) leads to a substantial performance drop, proving the necessity of combining both. Furthermore, our proposed hierarchical matching and score fusion mechanism shows clear superiority over simple feature concatenation. This quantitatively proves that our method not only effectively fuses multimodal information but also successfully avoids the core issue of introducing noise by directly combining high- and low-dimensional features, thus validating the rationality and efficiency of our architecture design.

Effectiveness of Ranking Block. The data in Table 10 shows that incorporating the ranking block leads to a noteworthy enhancement in object-level AUROC and object-level AUPR. However, there is only a minimal enhancement in point-level metrics. The ranking block is mainly used to reduce the significant influence of outliers on various anomaly score distributions. It plays an important role in balancing object-level anomaly scores, especially when the sample size is small. On the contrary, point-level anomaly scores contain a large number of samples, thereby reducing the impact of outliers. Thus, the ranking block has a limited impact on point-level metrics.

Effect of Different Distance Metrics. The use of Euclidean distance for both low-dimensional coordinates and high-dimensional semantic features warrants discussion. While Euclidean distance is a natural choice for geometric locality in 3D space, its applicability to high-dimensional embeddings can be debated. Our primary motivation for using Euclidean distance was to maintain a fair and direct comparison with the baseline method, Reg3D-AD, which employs the same metric. To further investigate this, we conducted additional experiments to evaluate the impact of different distance metrics for calculating the similarity of high-dimensional features. As shown in Table 13, we replaced the Euclidean distance with Cosine and Manhattan distances and re-evaluated the model's performance. The results indicate that while Euclidean distance yields the best performance in our framework, other common metrics also achieve competitive results, demonstrating the robustness of our approach.

Table 13
Performance of PointCore with different distance metrics for high-dimensional feature similarity.

Metric	Euclidean	Cosine	Manhattan
O-AUROC ↑	0.829	0.815	0.798
P-AUROC ↑	0.731	0.720	0.705
O-AUPR ↑	0.813	0.801	0.785
P-AUPR ↑	0.251	0.245	0.230

Trade-off Analysis of Local Optimization. To address concerns regarding the computational overhead of surface normal estimation and the Iterative Closest Point (ICP) algorithm, we provide a detailed breakdown of the inference time in Table 14. As observed, the *Normal Estimation* step—essential for the point-to-plane metric—consumes only 0.177s (4.14% of total time), while the *Local Optimization* itself takes 0.160s (3.73%). This combined overhead of approximately 0.34s is marginal when weighed against the substantial performance gains it unlocks: an increase in O-AUROC from 0.785 to 0.829 and a dramatic improvement in localization precision (P-AUPR) from 0.136 to 0.251 (see Table 10). Crucially, the overall efficiency of PointCore stems from the *PointMAE Anomaly Scoring* stage. Unlike the baseline Reg3D-AD (13.022s), which performs computationally expensive global nearest-neighbor searches directly in the high-dimensional (1154-D) feature space, our method utilizes the unified memory bank to perform a coordinate-guided search. This strategy restricts the semantic search space, keeping the feature scoring time to just 1.918s. Consequently, even with the addition of precise normal estimation and local registration, PointCore achieves state-of-the-art accuracy while maintaining an inference speed ($4.282s$) that is 3× faster than the baseline.

4.6. Robustness Analysis

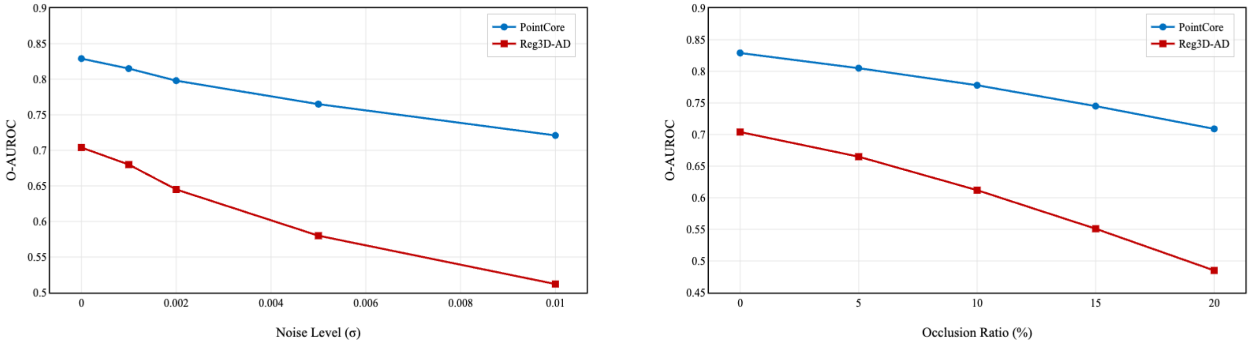
To further evaluate the stability and reliability of our proposed PointCore framework in practical applications, we conducted a series of robustness analysis experiments. Real-world point cloud data often suffers from sensor noise and partial occlusion due to scanning limitations or object obstruction. Therefore, it is crucial to assess how the model's performance is affected by these common data imperfections.

Experimental Design. We designed two sets of experiments to simulate these challenging scenarios.

- **Noise Robustness:** We added Gaussian noise with a mean of zero and varying standard deviations ($\sigma \in \{0.001, 0.002, 0.005, 0.01\}$) to the coordinates of the test point clouds. This simulates the random measurement errors typically produced by 3D scanners.
- **Occlusion Robustness:** We simulated partial occlusion by randomly selecting a point on the object's surface and removing all points within a spherical region around it. We tested this with different occlusion

Table 14
Detailed breakdown of inference time for PointCore (Total: 4.282s). The PointMAE anomaly scoring stage benefits significantly from our coordinate-guided search strategy.

Stage	Time (s)	Ratio (%)	Description
Global Registration	0.558	13.02%	Coarse alignment using FPFH features and RANSAC.
Normal Estimation	0.177	4.14%	Computing surface normals required for point-to-plane metric.
Local Optimization	0.160	3.73%	Fine-tuning alignment via point-to-plane ICP.
Feature Extraction	0.424	9.91%	Extracting deep representations using pre-trained PointMAE.
Coordinate Anomaly Scoring	0.840	19.61%	Calculating anomaly scores in the low-dimensional coordinate space.
PointMAE Anomaly Scoring	1.918	44.80%	Calculating anomaly scores in the high-dimensional feature space.
Miscellaneous	0.205	4.78%	Data preprocessing, memory allocation, and other overheads.
All	4.282	100%	



(a) Robustness to Gaussian Noise

(b) Robustness to Occlusion

Figure 8: Robustness analysis of PointCore and Reg3D-AD. (a) Performance (O-AUROC) under different levels of Gaussian noise. (b) Performance under different occlusion ratios. PointCore consistently demonstrates a more graceful performance degradation, indicating superior robustness.

proportions, removing 5%, 10%, 15%, and 20% of the total points.

We performed these tests on three representative datasets from Real3D-AD: **Candybar**, **Fish**, and **Seahorse**. We then averaged the O-AUROC scores and compared the performance degradation of PointCore against our main baseline, Reg3D-AD. **Results and Interpretation.** The results of our robustness analysis are visualized in Figure 8. As shown in Figure 8a, while the performance of both methods degrades with increasing noise levels, PointCore exhibits significantly higher resilience. Its O-AUROC score declines much more slowly than that of Reg3D-AD, maintaining a substantial performance margin even at high noise levels. We attribute this to our unified coordinate-semantic memory bank, where the initial search in the low-dimensional coordinate space effectively filters out minor noise-induced feature variations, leading to more stable semantic matching. Similarly, in the occlusion experiments shown in Figure 8b, PointCore demonstrates superior robustness. The performance drop is less pronounced as the occlusion ratio increases. This suggests that our model's ability to leverage both local and global features allows it to infer the overall structure and identify anomalies even when parts of the object are missing. In contrast, Reg3D-AD, which relies on separate feature banks, appears to be more sensitive to the loss of local information. These findings confirm that PointCore is not only accurate under ideal conditions but also more robust and practical for real-world deployment where data quality may be compromised.

5. Limitations and Future Work

While PointCore demonstrates state-of-the-art performance on datasets composed of rigid industrial objects, we acknowledge that its current architecture has certain limitations. The core of our methodology relies on a registration-based pipeline, which aligns test samples to a canonical template coordinate system. This fundamental assumption of a consistent geometric structure makes our approach highly effective for rigid objects, such as those found in industrial quality control, but less suitable for non-rigid or deformable objects (e.g., textiles, soft plastics, or organic materials).

The registration process would likely fail or produce unreliable alignments for objects that can bend, stretch, or change shape, leading to inaccurate anomaly detection. Therefore, the direct application of PointCore to such scenarios is not recommended without significant modifications.

Extending our framework to handle deformable objects presents a challenging but promising direction for future research. Potential avenues could include:

- **Part-based Models:** Instead of registering the entire object, one could learn to segment the object into semi-rigid parts and perform local registration and feature matching on a per-part basis.

- **Learning Registration-Invariant Features:** Developing deep learning models that can extract features invariant to non-rigid deformations, thus reducing the strict dependency on precise geometric alignment.

- **Canonical Shape Embedding:** Investigating methods to map deformed point clouds into a canonical, undeformed latent space before applying our memory-bank-based anomaly detection.

We believe that exploring these directions will significantly broaden the applicability of point cloud anomaly detection in more diverse and complex real-world scenarios.

6. Conclusion

This paper introduces a new unsupervised architecture for detecting anomalies in point clouds, called PointCore. The approach we use involves a single memory bank that combines local and global features to retain information from input point clouds at multiple scales. Extensive testing on the Real3D-AD dataset shows that our approach achieves higher recall rates and lower false-positive rates, making it ideal for accurate detection of defective samples in practical scenarios. Additionally, the suggested framework is effective because our integration of the local-global feature memory bank and the calculation of the multi-feature anomaly score based on coordinate distance significantly decrease computation expenses.

CRediT authorship contribution statement

Baozhu Zhao: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation. **Xiaohan Zhang:** Writing – review & editing, Supervision. **Jingfeng Guo:** Writing – review & editing, Supervision. **Qi Liu:** Writing – review & editing, Supervision, Resources, Project administration, Investigation, Funding acquisition.

References

Ahmed, A., Zakarya, M., Liu, X., Khan, R., Ali, A., Khan, A.A., 2022. Quality enhancement in a mm-wave multi-hop, multi-tier heterogeneous 5g network architecture. Telecommunication Systems 80, 169–187.

Ali, A., Naeem, H.Y., Sharafian, A., Qiu, L., Wu, Z., Bai, X., 2025a. Dynamic multi-graph spatio-temporal learning for citywide traffic flow prediction in transportation systems. Chaos, Solitons & Fractals 199, 116898.

Ali, A., Ullah, I., Singh, S.K., Jiang, W., Alturise, F., Bai, X., 2025b. Attention-driven graph convolutional networks for deadline-constrained virtual machine task allocation in edge computing. IEEE Transactions on Consumer Electronics .

Ali, R., Ali, A., Naeem, H.M.Y., Asad, M., Alsarhan, T., Heyat, B.B., 2024. A comprehensive survey of deep learning-based traffic flow prediction models for intelligent transportation systems. ICCK Transactions on Advanced Computing and Systems 1, 117–137.

Ali, R., Shen, Y., Huang, X., Zhang, J., Ali, A., 2017. Vmr: virtual machine replacement algorithm for qos and energy-awareness in cloud data centers, in: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), IEEE. pp. 230–233.

Asad, M., Azeem, W., Malik, A.A., Jiang, H., Ali, A., Yang, J., Liu, W., 2025. 3d-mmfn: Multi-level multimodal fusion network for 3d industrial image anomaly detection. Advanced Engineering Informatics 65, 103284.

Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., Steger, C., 2021. The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. International Journal of Computer Vision , 1038–1059URL: http://dx.doi.org/10.1007/s11263-020-01400-4, doi:10.1007/s11263-020-01400-4.

Blum, H., Sarlin, P.E., Nieto, J., Siegwart, R., Cadena, C., 2019. Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). URL: http://dx.doi.org/10.1109/iccvw.2019.00294, doi:10.1109/iccvw.2019.00294.

Cao, Y., Xu, X., Shen, W., 2024. Complementary pseudo multimodal feature for point cloud anomaly detection. Pattern Recognition 156, 110761.

Carrera, D., Manganini, F., Boracchi, G., Lanzarone, E., 2017. Defect detection in sem images of nanofibrous materials. IEEE Transactions on Industrial Informatics , 551–561URL: http://dx.doi.org/10.1109/tii.2016.2641472, doi:10.1109/tii.2016.2641472.

Chang, A., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F., 2015a. Shapenet: An information-rich 3d model repository. arXiv: Graphics,arXiv: Graphics .

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al., 2015b. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 .

Chen, K., Li, H., Li, C., Zhao, X., Wu, S., Duan, Y., Wang, J., 2022. An automatic defect detection system for petrochemical pipeline based on cycle-gan and yolo v5. Sensors 22, 7907.

Cheng, Y., Cao, Y., Xie, G., Lu, Z., Shen, W., 2024. Towards zero-shot point cloud anomaly detection: A multi-view projection framework. arXiv preprint arXiv:2409.13162 .

Chu, Y.M., Chieh, L., Hsieh, T.I., Chen, H.T., Liu, T.L., 2023. Shape-guided dual-memory learning for 3d anomaly detection .

Croitoru, F.A., Ristea, N.C., Dăscălescu, D., Ionescu, R.T., Khan, F.S., Shah, M., 2024. Lightning fast video anomaly detection via multi-scale adversarial distillation. Computer Vision and Image Understanding 247, 104074.

Defard, T., Setkov, A., Loesch, A., Audigier, R., 2020. Padim: a patch distribution modeling framework for anomaly detection and localization. Cornell University - arXiv,Cornell University - arXiv .

Deng, H., Li, X., 2022. Anomaly detection via reverse distillation from one-class embedding, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). URL: http://dx.doi.org/10.1109/cvpr52688.2022.00951, doi:10.1109/cvpr52688.2022.00951.

Du, M., Gu, S., Qin, Z., Xie, L., Wang, Z., Hu, Y., 2025. A generalized defect-data-free defect inspection method based on image reconstruction and anomaly detection. Neural Networks , 107662.

Ehret, T., Davy, A., Morel, J.M., Delbracio, M., 2019. Image anomalies: a review and synthesis of detection methods. Journal of Mathematical Imaging and Vision , 710–743URL: http://dx.doi.org/10.1007/s10851-019-00885-0, doi:10.1007/s10851-019-00885-0.

Fan, J., Ge, Y., Zhang, X., Wang, Z., Wu, H., Wu, J., 2024. Learning the feature distribution similarities for online time series anomaly detection. Neural Networks 180, 106638.

Fan, Y., Wen, G., Li, D., Qiu, S., Levine, M.D., Xiao, F., 2020. Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder. Computer Vision and Image Understanding 195, 102920.

Gudovskiy, D., Ishizaka, S., Kozuka, K., 2022. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows, in: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). URL: http://dx.doi.org/10.1109/wacv51458.2022.00188, doi:10.1109/wacv51458.2022.00188.

Hendrycks, D., Basart, S., Mazeika, M., Mostajabi, M., Steinhardt, J., Song, D., 2019. A benchmark for anomaly segmentation. .

Horwitz, E., Hoshen, Y., 2022. Back to the feature: Classical 3d features are (almost) all you need for 3d anomaly detection .

Hu, X., Li, Z., Luo, L., Karimi, H.R., Zhang, D., 2025. Dictionary trained attention constrained low rank and sparse autoencoder for hyperspectral anomaly detection. Neural Networks 181, 106797.

Li, G., He, P., Li, H., Zhang, F., 2023. Adversarial composite prediction of normal video dynamics for anomaly detection. Computer Vision and Image Understanding 232, 103686.

Li, H., Wang, Y., Wang, Y., Chen, J., 2025. A multi-memory-augmented network with a curvy metric method for video anomaly detection. Neural Networks 184, 106972.

Li, W., Xu, X., 2023. Towards scalable 3d anomaly detection and localization: A benchmark via 3d anomaly synthesis and a self-supervised learning network .

Li, W., Xu, X., Gu, Y., Zheng, B., Gao, S., Wu, Y., 2024. 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, pp. 22207–22216.

Liu, J., Xie, G., Chen, R., Li, X., Wang, J., Liu, Y., Wang, C., Zheng, F., 2023. Real3d-ad: A dataset of point cloud anomaly detection .

Lu, T., Wang, Z., Shen, Y., Shao, X., Tang, Y., 2023. Defvae: A defect detection method for catenary devices based on variational autoencoder. IEEE Transactions on Instrumentation and Measurement .

Ma, X., Wu, J., Liu, W., 2025. Sac-bl: A hypothesis testing framework for unsupervised visual anomaly detection and location. Neural Networks , 107147.

Masuda, M., Hachiuma, R., Fujii, R., Saito, H., Sekikawa, Y., 2021a. Toward unsupervised 3d point cloud anomaly detection using variational autoencoder, in: 2021 IEEE International Conference on Image Processing (ICIP). URL: http://dx.doi.org/10.1109/icip42928.2021.9506795, doi:10.1109/icip42928.2021.9506795.

Masuda, M., Hachiuma, R., Fujii, R., Saito, H., Sekikawa, Y., 2021b. Toward unsupervised 3d point cloud anomaly detection using variational autoencoder, in: 2021 IEEE International Conference on Image Processing (ICIP), IEEE. pp. 3118–3122.

Mei, S., Wang, Y., Wen, G., 2018. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. Sensors 18, 1064.

Pang, G., Shen, C., Cao, L., Hengel, A.V.D., 2022a. Deep learning for anomaly detection: A review. ACM Computing Surveys , 1–38URL: http://dx.doi.org/10.1145/3439950, doi:10.1145/3439950.

Pang, Y., Wang, W., Tay, F., Liu, W., Tian, Y., Yuan, L., 2022b. Masked autoencoders for point cloud self-supervised learning .

Qin, J., Gu, C., Yu, J., Zhang, C., 2023. Teacher–student network for 3d point cloud anomaly detection with few normal samples. Expert Systems with Applications 228, 120371.

Roth, K., Pemula, L., Zepeda, J., Scholkopf, B., Brox, T., Gehler, P., 2022. Towards total recall in industrial anomaly detection, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). URL: http://dx.doi.org/10.1109/cvpr52688.2022.01392, doi:10.1109/cvpr52688.2022.01392.

Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the icp algorithm, in: Proceedings Third International Conference on 3-D Digital Imaging and Modeling, pp. 145–152. doi:10.1109/IM.2001.924423.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (fpfh) for 3d registration, in: 2009 IEEE International Conference on Robotics and Automation, pp. 3212–3217. doi:10.1109/ROBOT.2009.5152473.

Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M., 2008. Aligning point cloud views using persistent feature histograms, in: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3384–3391. doi:10.1109/IROS.2008.4650967.

Schwartz, E., Arbelle, A., Karlinsky, L., Harary, S., Scheidegger, F., Doveh, S., Giryes, R., 2024. Maeday: Mae for few-and zero-shot anomaly-detection. Computer Vision and Image Understanding 241, 103958.

Sharafian, A., Naeem, H.Y., Ullah, I., Ali, A., Qiu, L., Bai, X., 2025. Resilience to deception attacks in consensus tracking control of incommensurate fractional-order power systems via adaptive rbf neural network. Expert Systems with Applications , 127763.

Song, K., Yan, Y., 2013. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Applied Surface Science , 858–864URL: http://dx.doi.org/10.1016/j.apsusc.2013.09.002, doi:10.1016/j.apsusc.2013.09.002.

Wang, Y., Peng, J., Zhang, J., Yi, R., Wang, Y., Wang, C., 2023. Multimodal industrial anomaly detection via hybrid fusion .

Wang, Z., Gu, X., Gu, X., Hu, J., 2024. Enhancing video anomaly detection with learnable memory network: A new approach to memory-based auto-encoders. Computer Vision and Image Understanding 241, 103946.

Xie, G., Wang, J., Liu, J., Lyu, J., Liu, Y., Wang, C., Zheng, F., Jin, Y., 2023. Im-iad: Industrial image anomaly detection benchmark in manufacturing .

Yang, L., Guo, W., 2016. Greedy local-set based sampling and reconstruction for band-limited graph signals, in: 2016 23rd International Conference on Telecommunications (ICT), pp. 1–5. doi:10.1109/ICT.2016.7500429.

Yu, J., Zheng, Y., Wang, X., Li, W., Wu, Y., Zhao, R., Liwei, W., 2021. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. Cornell University - arXiv,Cornell University - arXiv .

Zavrtanik, V., Kristan, M., Skočaj, D., 2021. Draem – a discriminatively trained reconstruction embedding for surface anomaly detection. International Conference on Computer Vision,International Conference on Computer Vision .

Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V., 2021. Point transformer., in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). URL: http://dx.doi.org/10.1109/iccv48922.2021.01595, doi:10.1109/iccv48922.2021.01595.

Zhao, Z., Li, B., Dong, R., Zhao, P., 2018. A surface defect detection method based on positive samples, in: PRICAI 2018: Trends in Artificial Intelligence: 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28–31, 2018, Proceedings, Part II

15, Springer. pp. 473–481.

Zhou, Z., Wang, L., Fang, N., Wang, Z., Qiu, L., Zhang, S., 2025. R3d-ad: Reconstruction via diffusion for 3d anomaly detection, in: European Conference on Computer Vision, Springer. pp. 91–107.

Zhu, H., Xie, G., Hou, C., Dai, T., Gao, C., Wang, J., Shen, L., 2024. Towards high-resolution 3d anomaly detection via group-level feature contrastive learning, in: Proceedings of the 32nd ACM International Conference on Multimedia, pp. 4680–4689.