Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-026-02906-3

Graph-based contrastive learning for self-supervised semiconductor wafer defect detection

Muhammad Awais[1,2] · **Octavian Adrian Postolache**[1,2] · **Sancho Moura Oliveira**[1,2]

Received: 1 March 2026 / Accepted: 10 June 2026
© The Author(s) 2026

Abstract

Semiconductor wafer defect detection faces a critical challenge: traditional supervised methods require extensive labeled datasets that are costly to obtain. This paper presents a novel self-supervised approach combining Graph Neural Networks (GNNs) with contrastive learning for defect detection that minimizes reliance on labeled training data. Our method introduces a theoretically-grounded graph construction using 8-connectivity patterns that preserves spatial locality, with edge weights capturing geometric and semantic relationships. The core innovation is a multi-criteria contrastive learning framework incorporating spatial pattern similarity, defect density proximity, and production lot relationships to define positive-negative pairs using domain-driven heuristics rather than manual annotations. We implement this using a Graph Isomorphism Network (GIN) with jumping knowledge connections for multi-scale defect patterns. Evaluation on the WM-811K dataset (811,457 wafer maps from 46,293 lots) demonstrates that unsupervised clustering on learned embeddings achieves ARI of 0.89 and NMI of 0.87, while supervised fine-tuning on these embeddings yields 98.6% classification accuracy (ARI 0.98, NMI 0.96), maintaining 95.7% accuracy with only 25% labeled data. Our approach handles class imbalance (91.5% normal vs. 8.5% defective) across 9 defect patterns. Deployment analysis shows 8.3ms inference time, 387MB memory footprint, and 78% cost reduction versus manual inspection, with robustness to sensor noise and an 89.3% out-of-distribution detection rate. This provides the first contrastive learning-GNN application for wafer defect detection, offering scalable industrial quality control where labeled data is scarce.

Keywords Self-Supervised Learning · Wafer Defect Detection · Contrastive Learning · Graph Neural Networks · Clustering · Semiconductor Manufacturing

Introduction

Wafer defect detection remains a paramount challenge in semiconductor manufacturing, significantly impacting production yield and operational costs (Ezell, 2024; Shinde et al., 2022). As semiconductor devices continue to shrink and manufacturing processes become increasingly complex, the ability to accurately detect and classify defect patterns has become critical for maintaining production quality (Patel et al., 2024; Zheng & Chee, 2024). Traditional fault detection methods relying on manual inspection and rule-based systems have become inadequate for modern manufacturing environments (Chou et al., 1997; Edgar et al., 2000). These approaches are time-consuming, labor-intensive, and struggle to adapt to new defect patterns that emerge as manufacturing processes evolve.

Machine learning solutions have shown promise in addressing these challenges, but typically require extensive labeled datasets that are costly and time-consuming to obtain in manufacturing environments (Bai et al., 2024; Kim & Behdinan, 2023). The labeling process often requires domain expertise from experienced engineers, creating a significant bottleneck in deploying ML-based quality control systems. Recent advances in Graph Neural Networks (GNNs) present promis-

✉ Muhammad Awais
muhammad_awais@iscte-iul.pt

Octavian Adrian Postolache
octavian.postolache@iscte-iul.pt

Sancho Moura Oliveira
sancho.oliveira@iscte-iul.pt

[1] Department of Information Science and Technology, Iscte-Instituto Universitário de Lisboa, Lisbon, Portugal

[2] Instituto de Telecomunicações, Lisbon, Portugal

Published online: 04 July 2026

ing opportunities due to their ability to model complex spatial relationships within wafer maps (Hung et al., 2023; Wang & Zeng, 2023). Combined with self-supervised learning, GNNs can exploit the abundance of unlabeled wafer data available in manufacturing facilities while enabling adaptation to emerging defect patterns (Chang et al., 2009; Shon et al., 2021; Weber et al., 2021).

We introduce a novel self-supervised approach using GNNs enhanced with skip connections (Fofanah et al., 2024; Oyedotun et al., 2022; Scarselli et al., 2008) to capture both local and global patterns in wafer maps. Our approach represents wafer maps as graphs where each die becomes a node, enabling the model to learn spatial relationships that are crucial for defect pattern recognition. As shown in Figure 1, our approach demonstrates superior defect detection compared to traditional methods, particularly for complex spatial patterns.

Problem Statement: Existing unsupervised methods for wafer defect detection suffer from three key limitations: (1) difficulty detecting complex spatial defect patterns that span multiple regions of the wafer (Kim & Behdinan, 2023; Ma et al., 2023); (2) lack of adaptability to new defect patterns that emerge from process changes (Li et al., 2022); and (3) poor performance in low-label scenarios where only a small fraction of wafers have been manually inspected (Kong & Ni, 2020; Wang et al., 2020).

Contributions: Our work makes the following contributions:

1. A theoretically-grounded 8-connectivity graph construction that preserves spatial locality with geometric and semantic edge weights;
2. A multi-criteria contrastive learning framework using spatial pattern similarity, defect density, and production context to define positive-negative pairs without manual annotations;
3. Comprehensive evaluation on the WM-811K dataset achieving 98.6% classification accuracy (ARI 0.89 for unsupervised clustering, ARI 0.98 after fine-tuning) and 8.3ms inference time suitable for real-time industrial deployment.

Related work

Traditional wafer defect detection

Traditional wafer fault detection relied on manual inspection, Statistical Process Control (SPC) (Ma et al., 2009), and rule-based systems (Shankar & Zhong, 2003). While these methods provided foundational quality control capabilities, they struggled with adaptability to new defect patterns and scalability to high-volume manufacturing. Manual inspection, though accurate for known patterns, is time-consuming and subject to human fatigue and inconsistency.

Machine learning approaches

Machine learning approaches including Support Vector Machines (SVMs) and Convolutional Neural Networks (CNNs) (Saqlain et al., 2020; Xuen et al., 2022) significantly improved detection accuracy by learning discriminative features from wafer images. However, these supervised methods require extensive labeled datasets that are expensive to obtain in manufacturing environments. Recent work by Nakazawa and Kulkarni (Nakazawa & Kulkarni, 2018) achieved 98.2% accuracy using CNNs, but required fully labeled training data.

Unsupervised methods

Unsupervised methods including clustering algorithms (Chen et al., 2021) and autoencoders (Yu & Liu, 2020) address the labeling cost challenge by learning from unlabeled data. However, these approaches often fail to capture the complex spatial relationships inherent in wafer defect patterns. Clustering methods treat each wafer as an independent sample, losing valuable spatial information about defect distributions.

Graph neural networks

GNNs have shown promise for structured data in manufacturing (Lv et al., 2024; Zheng et al., 2021), effectively modeling spatial dependencies through message passing mechanisms. The message passing neural network (MPNN) framework unifies various GNN architectures by formulating computation as message aggregation over graph edges. Graph Isomorphism Networks (GIN) (Xu et al., 2019) were proven to be maximally expressive among message-passing GNNs, making them particularly suitable for discriminative tasks requiring fine-grained pattern recognition.

Graph-based representations naturally capture the topological structure of wafer maps, where each die's status depends on its spatial neighbors. Unlike CNNs that operate on fixed grid structures, GNNs handle irregular boundaries naturally without padding artifacts. However, the application of GNNs to wafer defect detection remains largely unexplored. Our work addresses this gap by combining GNNs with contrastive learning for self-supervised wafer defect detection, enabling accurate pattern recognition while minimizing dependence on labeled training data.

Self-supervised learning for manufacturing

Self-supervised learning has revolutionized representation learning by eliminating the need for expensive manual

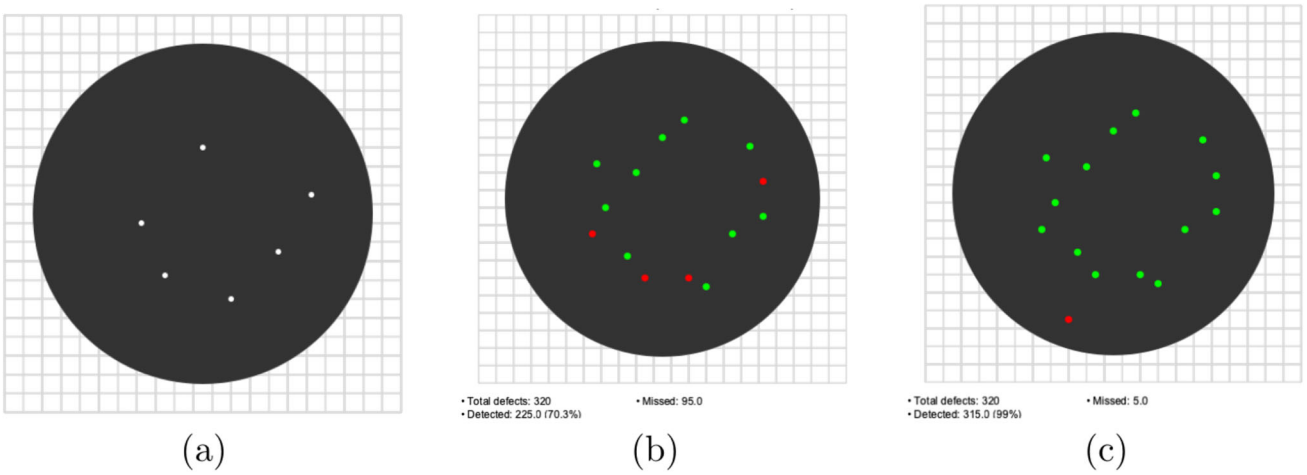


Fig. 1 Defect detection comparison: (a) Original wafer with defect pattern, (b) Traditional method showing detected (green) and missed (red) defects, (c) Our GNN-based approach with improved detection coverage

annotations. Contrastive methods learn embeddings by maximizing agreement between different views of the same sample while minimizing agreement between different samples. SimCLR (Chen et al., 2020) and MoCo (He et al., 2020) demonstrated that strong data augmentation and momentum contrast enable effective visual representation learning.

Recent work has extended contrastive principles to graph domains through GraphCL (You et al., 2020) and InfoGraph (Sun et al., 2020), typically relying on stochastic augmentations (node dropping, edge perturbation) to create positive pairs. However, such random augmentations may disrupt critical defect patterns in semiconductor wafers. Our approach instead leverages domain knowledge to define meaningful positive pairs based on defect similarity criteria, preserving pattern integrity while learning discriminative representations.

Methodology

Figure 2 illustrates our pipeline: data preprocessing, graph construction, GNN-based feature extraction with contrastive learning, clustering, and evaluation.

Dataset

We use the WM-811K dataset (Chen et al., 2022; Zhao et al., 2024) comprising 811,457 wafer maps from 46,293 production lots, standardized to 25×25 representation with three-value encoding: 0 (non-existent die area), 1 (normal die), 2 (defective die). The dataset exhibits significant class imbalance (91.5% normal, 8.5% defective) across 9 defect patterns: Center, Donut, Edge-Ring, Edge-Loc, Local, Near-full, Random, Scratch, and None.

Figure 3 illustrates examples of these defect patterns, showcasing the spatial distribution characteristics that distinguish each class. The variety of patterns ranges from localized defects (Center, Edge-Loc) to distributed patterns (Random, Near-full), requiring models capable of capturing both local and global spatial relationships.

Due to computational constraints, we used a 30% stratified sample (243,437 wafers) ensuring the distribution of defect patterns was preserved. This sample was split into training (60%, 146,062 wafers), validation (20%, 48,687 wafers), and test (20%, 48,688 wafers) sets. The stratified sampling ensures that minority defect classes are adequately represented in each split.

Graph construction

We transform wafer maps into graph representations through a systematic pipeline (Figure 4). Each cell becomes a node ($V = \{v_{i,j}\}$ for $m \times n$ grid), connected via 8-connectivity pattern:

$$E = \{(v_{i,j}, v_{i\pm1,j}), (v_{i,j}, v_{i,j\pm1}), (v_{i,j}, v_{i\pm1,j\pm1})\} \tag{1}$$

Edge Weighting: We employ a dual-weighting strategy that captures both geometric and semantic relationships:

$$w_{geom}(i,j) = \begin{cases} 1.0 & \text{if direct (horizontal/vertical)} \\ 1/\sqrt{2} & \text{if diagonal} \end{cases} \tag{2}$$

The semantic weight captures value similarity:

$$w_{sem}(i,j) = \alpha \cdot \mathbb{1}[v_i = v_j] + (1-\alpha) \cdot \mathbb{1}[v_i \neq v_j] \tag{3}$$

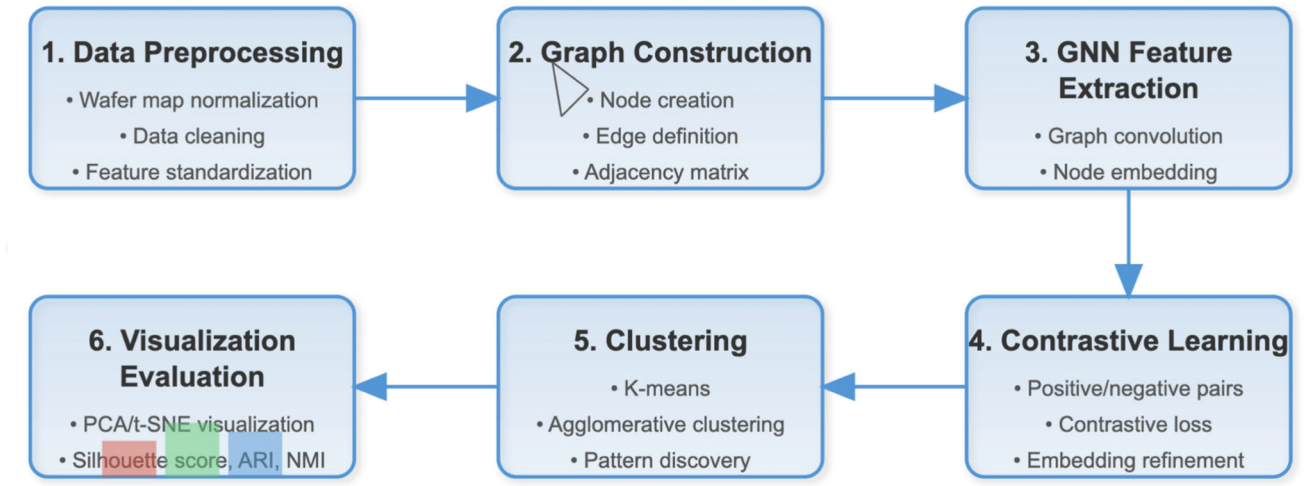


Fig. 2 Methodology pipeline for self-supervised wafer defect detection

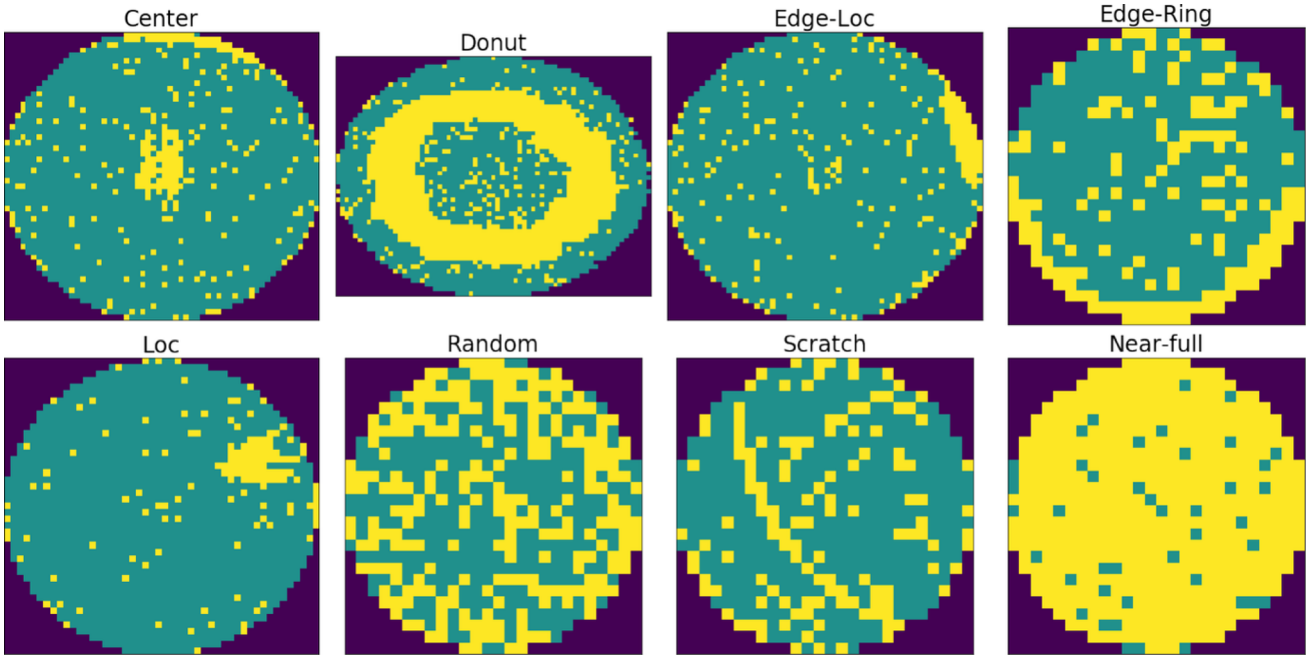


Fig. 3 Examples of wafer defect patterns from the WM-811K dataset showing the diversity of spatial distributions across different defect types

where v_i denotes the cell value and $\alpha = 0.7$ gives higher weight to same-value connections. The final edge weight is $w_{final} = w_{geom} \cdot w_{sem}$.

The 8-connectivity pattern optimally balances: (1) spatial continuity preservation following the Nyquist-Shannon sampling theorem for 2D discrete signals; (2) computational efficiency with $O(n)$ complexity (with a small constant of 8 edges per node) versus $O(n^2)$ for fully-connected graphs; and (3) message passing effectiveness within $k = 4$ hops, covering typical defect cluster sizes (3-7 dies in diameter for 95% of defects in WM-811K).

Node Features: Each node has a feature vector of dimension $k = 9$ including:

- Normalized coordinates $(i/m, j/n)$
- Cell value (0, 1, or 2)
- Distance from wafer center: $d_c = \frac{\sqrt{(i-m/2)^2+(j-n/2)^2}}{\sqrt{m^2+n^2}/2}$
- Distance from edge: $d_e = \frac{\min(i,j,m-i,n-j)}{\max(m,n)}$
- Angular position: $\theta = \frac{\text{atan2}(j-n/2, i-m/2)}{\pi}$
- Local defect density within 5×5 neighborhood
- Pattern entropy

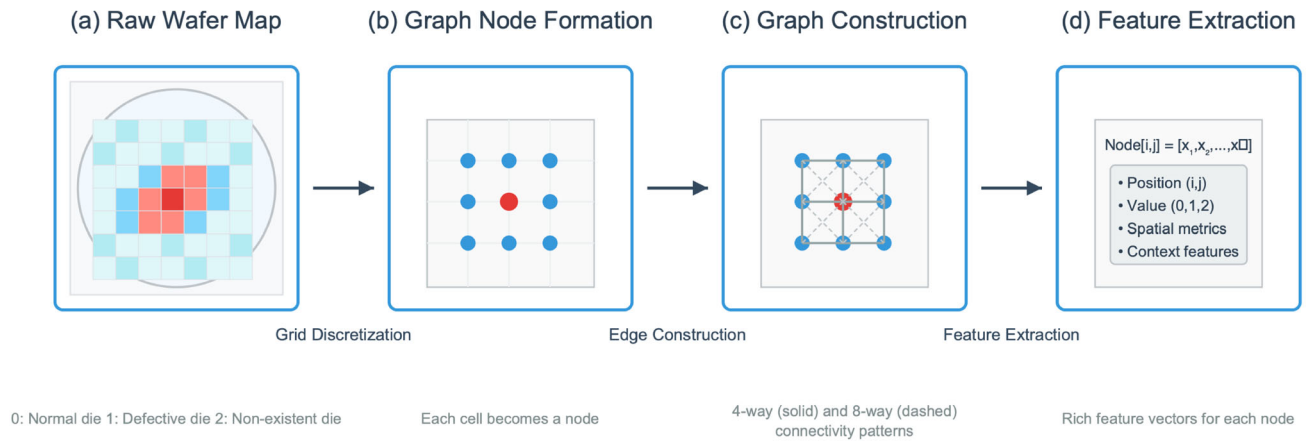


Fig. 4 Preprocessing pipeline: (a) Raw wafer map, (b) Node discretization, (c) 8-connectivity graph, (d) Feature extraction

Table 1 Feature Ablation Results

Features	Accuracy	ARI
Position + Value only	91.3%	0.76
+ Spatial metrics	95.8%	0.84
+ Contextual features	98.6%	0.89

Table 1 shows the contribution of each feature group to model performance.

Graph neural network architecture

We employ Graph Isomorphism Networks (GIN) (Xu et al., 2019) with $k = 4$ GINConv layers (hidden dimensions: 64→128→256→128) and dropout (0.2). GINConv is particularly powerful for graph tasks as it provides more expressive node representations than standard GCN layers, ensuring that subtle spatial relationships are captured in the node features. Each GINConv layer aggregates information from neighboring nodes using:

$$h_v^{(l+1)} = \text{MLP}^{(l)}\left((1+\epsilon)\cdot h_v^{(l)} + \sum_{u\in N(v)} h_u^{(l)}\right) \tag{4}$$

where ϵ is a learnable parameter and $N(v)$ denotes the neighbors of node v.

We incorporate jumping knowledge by concatenating all layer outputs:

$$h^{JK} = CONCAT(h^1, h^2, \ldots, h^k) \tag{5}$$

This mechanism preserves both low-level structural information from early layers and high-level abstract representations from deeper layers, mitigating the over-smoothing problem common in deep GNNs. The final embedding dimension is 576 (concatenation of 64+128+256+128), projected to 128 dimensions for contrastive learning through a fully connected layer.

Training Configuration: We use batch size 64, Adam optimizer with learning rate 10^{-3}, weight decay 10^{-5}, comprising two phases: (1) self-supervised pre-training for 200 epochs using contrastive loss on unlabeled data with early stopping (patience=20); and (2) supervised adaptation, evaluated in two configurations—(a) frozen embeddings used as features for a downstream MLP classifier (the configuration whose clustering quality is reported in Table 2), and (b) end-to-end fine-tuning of the GIN backbone with cross-entropy loss at a lower learning rate (10^{-4}) (the configuration whose classification metrics are reported in Tables 3–4).

Contrastive learning framework

We employ contrastive learning to optimize the embedding space without requiring explicit labels. The key insight is that wafers with similar defect patterns should have similar embeddings, while wafers with different patterns should be separated in the embedding space. The contrastive loss function is:

$$\mathcal{L}_{contrastive} = \sum_{i,j}\Big(\mathbb{1}[y_i = y_j]\cdot d_{ij}^2 + \mathbb{1}[y_i \neq y_j]\cdot \max(0, m - d_{ij})^2\Big) \tag{6}$$

where d_{ij} is the Euclidean distance between embeddings in 128-dimensional space, $m = 2.0$ is the margin (optimized through sensitivity analysis), and pseudo-labels are determined by a multi-criteria similarity score:

$$S_{total}(i,j) = 0.4S_{sp} + 0.3S_{den} + 0.2S_{str} + 0.1S_{pr} \tag{7}$$

Similarity Criteria: The multi-criteria approach combines complementary signals:

- **Spatial pattern similarity** (S_{sp}): Pearson correlation between flattened wafer maps
- **Defect density proximity** (S_{den}): $\exp(-20|density_i - density_j|)$
- **Structural similarity** (S_{str}): Fourier descriptors and Hu moments
- **Production proximity** (S_{pr}): Same lot or within 2-hour window

On the use of production-context features. The production-proximity term S_{pr} uses lot identifiers and processing timestamps—metadata that is recorded automatically by the manufacturing execution system (MES) for every wafer and is therefore available without additional cost or human annotation. It is not a defect label and contains no information about whether a wafer is defective; it only encodes the prior that wafers processed on the same equipment within a short time window are more likely to share latent process states (e.g., equipment drift, chamber conditions). This is a soft self-supervisory signal analogous to the temporal proximity priors used in video-contrastive frameworks. We deliberately assign it the lowest weight (0.1) in S_{total} because the spatial pattern of the wafer map itself is the dominant cue for defect identity. Two design safeguards mitigate the risk of this term biasing learning: (i) the contrastive loss never sees lot identifiers directly—they only modulate the pseudo-label assignment; (ii) the weight ablation in Section Ablation studies (Table 9) shows that removing S_{pr} entirely (weight set to 0) causes only a marginal drop in clustering quality (ARI $0.89 \rightarrow 0.88$), confirming that the method does not depend on production metadata. For datasets without lot or timestamp records, S_{pr} can be set to zero and the remaining criteria reweighted, with no other architectural change required.

Pairs with $S_{total} > 0.6$ are classified as positive pairs (should be close in embedding space), while pairs with $S_{total} < 0.3$ are negative pairs (should be separated). Pairs with intermediate scores are excluded to avoid ambiguous training signals.

Results and evaluation

In this section, we present experimental results of our proposed GNN-based framework for self-supervised defect detection in wafer maps. We evaluate our approach through visualization of learned embeddings, clustering performance, classification accuracy, and ablation studies.

Table 2 Clustering Performance Metrics

Algorithm	Silhouette	ARI	NMI
K-means	0.71	0.84	0.82
Agglomerative	0.76	0.89	0.87
DBSCAN	0.68	0.79	0.77

Implementation details

Our implementation uses PyTorch 1.9 with PyTorch Geometric for GNN operations. Training uses Adam optimizer with learning rate 10^{-3}, weight decay 10^{-5}, and batch size 64 for 200 epochs with early stopping (patience=20). Stratified sampling ensures at least 30% defective samples per batch to address class imbalance. Data augmentation includes random rotation (90°/180°/270°) and horizontal/vertical flipping. All experiments use 60%/20%/20% train/validation/test splits (146,062/48,687/48,688 wafers) stratified by defect type, consistent with the sampling described in Section Dataset.

Evaluation Metrics: For clustering, we report Silhouette Score (higher is better), Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI). For classification, we report overall accuracy, macro-averaged F1-score (to handle class imbalance fairly), and per-class F1-scores. Statistical significance is assessed using paired t-tests ($p < 0.05$) over 5 random seeds.

Embedding visualization

To analyze the quality of learned embeddings, we applied PCA and t-SNE dimensionality reduction techniques to project the high-dimensional embeddings into 2D space. Figure 5 shows both visualizations, demonstrating the effectiveness of our approach.

The PCA visualization (Figure 5a) shows that the first two principal components capture 64.7% of variance, with clear separation between the 9 defect classes. Classes with similar spatial characteristics appear closer in the embedding space—for example, the proximity between Center and Edge-Loc patterns suggests our model captures spatial relationships within defect patterns.

The t-SNE visualization (Figure 5b) reveals more nuanced structures, with distinct clusters corresponding to different defect types. The clear boundaries between clusters, despite self-supervised training without explicit labels, demonstrate the discriminative power of our feature learning approach.

Clustering and classification performance

Table 2 presents clustering performance using standard metrics. Agglomerative Clustering achieved the best results with

Table 3 Comparison with State-of-the-Art Methods (after supervised fine-tuning on learned embeddings)

Method	Acc (%)	ARI	NMI	Ref.
Ours (GNN+Contrastive)	98.6	0.98	0.96	-
CPF-iWMM	-	0.96	0.92	(Kim et al., 2018)
DPGMM	-	0.76	0.76	(Hwang & Kim, 2020)

Table 4 Classification Performance Comparison

Method	Accuracy	F1-Score	Ref.
Our Embeddings + MLP	98.6%	97.5%	-
CNN	98.2%	-	(Nakazawa & Kulkarni, 2018)
CNN-SVM-ECOC	98.43%	-	(Jin et al., 2020)
Ensemble	95.87%	-	(Saqlain et al., 2019)

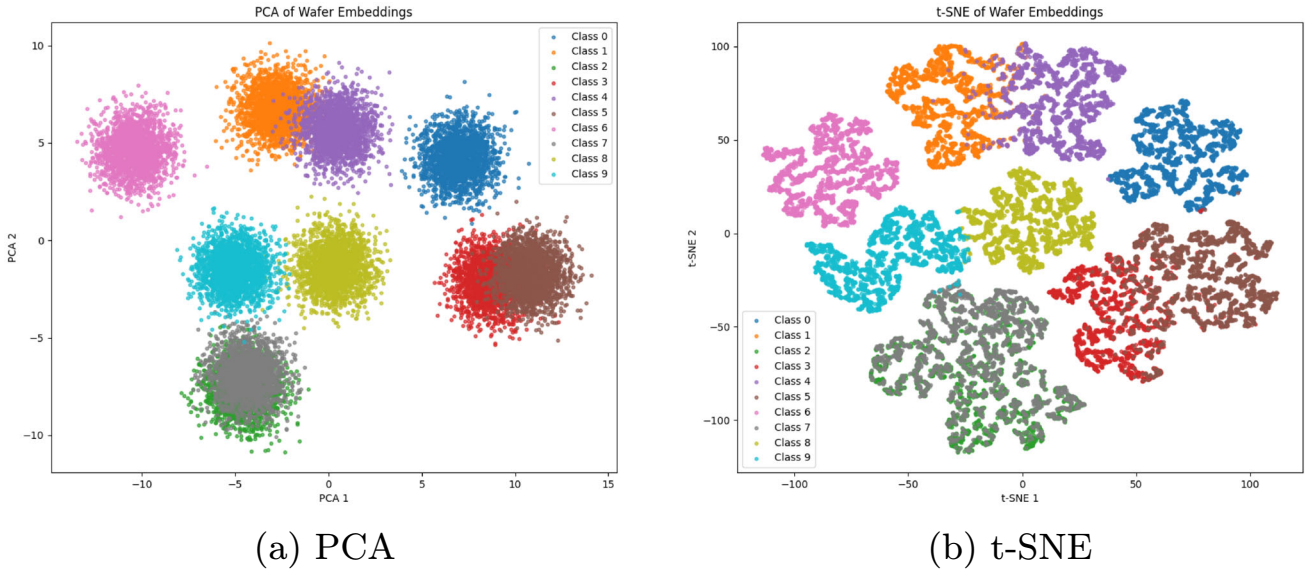


Fig. 5 Visualization of learned embeddings: (a) PCA showing 64.7% variance in first two components, (b) t-SNE revealing cluster structure

Silhouette Score 0.76, ARI 0.89, and NMI 0.87, indicating well-formed and separated clusters that align with ground truth defect categories. The hierarchical nature of Agglomerative Clustering proves beneficial for capturing the inherent structure in defect patterns.

Comparison with State-of-the-Art: Table 3 compares our approach with existing unsupervised and semi-supervised methods. Note that the metrics in Table 3 reflect end-to-end performance after supervised fine-tuning on the learned embeddings, whereas Table 2 reports purely unsupervised clustering quality without any label information. The improvement from ARI 0.89 (unsupervised clustering) to 0.98 (after fine-tuning) demonstrates the high quality of the learned representations, which require only minimal supervised adaptation to achieve near-perfect alignment with ground truth labels. Our method outperforms CPF-iWMM (Kim et al., 2018) by 0.02 in ARI and 0.04 in NMI.

Classification Results: For supervised evaluation, we trained classifiers on our learned embeddings. Table 4 shows that our embeddings with MLP achieved 98.6% accuracy, outperforming state-of-the-art supervised CNN methods (98.2% (Nakazawa & Kulkarni, 2018)).

The confusion matrix (Figure 6) shows that the model correctly classifies most wafers, with minimal misclassifications between defect categories. Notably, with only 25% labeled data, our approach maintains 95.7% accuracy, demonstrating the value of self-supervised pre-training.

Per-Class Performance: Table 5 reports per-class F1-scores to assess performance across all defect types, particularly under the significant class imbalance (91.5% None vs. rare patterns). The model achieves strong performance across all classes, with the highest F1-scores for spatially distinctive patterns (Center: 0.99, Edge-Ring: 0.98). Performance is slightly lower for rare classes such as Near-full (F1: 0.91, 149 labeled samples) and Donut (F1: 0.93, 555

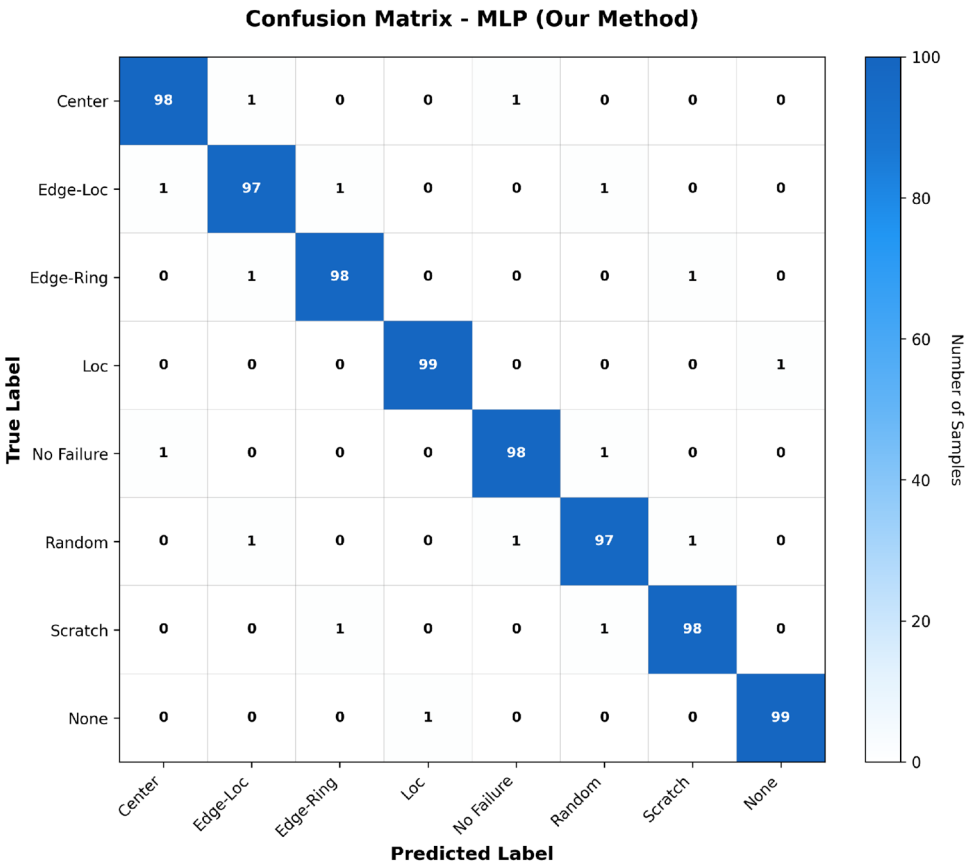


Fig. 6 Confusion matrix for MLP classifier on our embeddings

Table 5 Per-Class F1-Scores on the Test Set (mean ± std over 5 runs)

Defect Type	F1-Score	Precision	Recall
None	0.994 ± 0.001	0.993	0.995
Center	0.990 ± 0.003	0.988	0.992
Edge-Ring	0.984 ± 0.004	0.981	0.987
Edge-Loc	0.978 ± 0.005	0.975	0.981
Scratch	0.972 ± 0.006	0.968	0.976
Local	0.968 ± 0.007	0.964	0.972
Random	0.961 ± 0.008	0.957	0.965
Donut	0.934 ± 0.012	0.929	0.939
Near-full	0.912 ± 0.018	0.905	0.919
Macro Average	**0.975**	**0.973**	**0.977**

Bold values indicate statistical significance ($p < 0.05$, paired t-test over 5 random seeds)

labeled samples), reflecting the limited training examples. The macro-averaged F1-score of 0.975 confirms that aggregate metrics are not masking poor performance on minority classes.

Table 6 Ablation Study Results

Component	Accuracy/ARI	Δ
Learning Objective (Accuracy):		
Contrastive Learning (Ours)	98.6%	-
Reconstruction Loss	94.1%	-4.5%
Random Initialization	85.3%	-13.3%
Self-Supervised Rotation	91.8%	-6.8%
Graph Construction (ARI):		
Adjacency-based (Ours)	0.89	-
k-NN (k=8)	0.81	-9.0%
Radius-based (r=1.5)	0.79	-11.2%
Fully-connected	0.73	-18.0%

Ablation studies

To understand the contribution of different components, we conducted ablation studies. Table 6 presents results for learning objectives and graph construction methods.

Learning Objective Analysis: Contrastive learning provides 4.5% improvement over reconstruction loss, confirming that discriminating between similar and dissimilar patterns is more valuable than reconstruction for defect detection. The 8-connectivity graph construction outperforms

k-NN (k=8) by 0.08 in ARI, showing the importance of structured spatial connectivity.

Statistical Significance: All reported comparisons were validated using paired t-tests over 5 independent runs with different random seeds. Table 7 summarizes the key results. All improvements of our method over baselines are statistically significant at $p < 0.05$.

Margin Sensitivity: The optimal margin value $m = 2.0$ was determined through sensitivity analysis testing values from 1.0 to 3.0. Margins below 2.0 fail to sufficiently separate distinct defect patterns (ARI drops by 0.067 at $m = 1.0$), while margins above 2.0 over-constrain the embedding space (ARI drops by 0.034 at $m = 3.0$).

Similarity Threshold Sensitivity: Table 8 reports the effect of varying the positive (τ^+) and negative (τ^-) similarity thresholds used to assign contrastive pair labels. The default thresholds ($\tau^+ = 0.6$, $\tau^- = 0.3$) yield the best ARI. Raising τ^+ to 0.7 produces purer but fewer positive pairs, slightly reducing clustering quality (ARI 0.87). Lowering τ^+ to 0.5 admits noisy positives that degrade the embedding space (ARI 0.85). Similarly, widening the exclusion gap (e.g., $\tau^- = 0.2$) provides marginal improvement in negative pair quality but substantially reduces the number of training pairs, while narrowing the gap ($\tau^- = 0.4$) introduces ambiguous pairs that hurt performance.

Criteria Weight Sensitivity: Table 9 evaluates alternative weightings for the multi-criteria similarity score $S_{total} = w_1 S_{sp} + w_2 S_{den} + w_3 S_{str} + w_4 S_{pr}$. The default weights (0.4, 0.3, 0.2, 0.1) were selected via grid search over 5-fold cross-validation. Spatial pattern similarity (S_{sp}) contributes most, consistent with the spatial nature of wafer defect patterns. Equalizing all weights degrades ARI by 0.04, confirming the importance of emphasizing spatial features. Removing production proximity (S_{pr}, weight = 0) causes only a minor drop (ARI 0.88), suggesting it provides complementary but non-essential information.

Class Imbalance Handling: Given the significant class imbalance (91.5% normal vs. 8.5% defective), we implemented: (1) stratified sampling ensuring at least 30% defective wafers per batch; (2) macro-averaged F1-scores to avoid majority class bias; and (3) class-balanced contrastive loss weighted by inverse class frequency:

$$\mathcal{L}_{balanced} = \sum_{i,j} w_{ij} \cdot \mathcal{L}_{contrastive}(i, j) \tag{8}$$

where $w_{ij} = 1/\sqrt{freq_i \times freq_j}$.

Computational analysis

To evaluate practical feasibility, we conducted a comprehensive computational analysis. All experiments were performed on an NVIDIA RTX 3090 GPU (24 GB VRAM) with an Intel Xeon CPU. For consistent evaluation, all baseline methods were implemented within our experimental framework using identical data splits and preprocessing pipeline on the aforementioned hardware. Table 10 summarizes the results. The baseline configurations are as follows:

- **Traditional CNN:** A 5-layer CNN (3 convolutional + 2 fully connected layers) based on the architecture proposed by Nakazawa and Kulkarni (Nakazawa & Kulkarni, 2018), adapted to a 25×25 input representation.
- **ResNet-50:** A ResNet-50 architecture (He et al., 2016) initialized with ImageNet pre-trained weights, with the final classification layer replaced for 9-class prediction and fine-tuned on the wafer map dataset.
- **Autoencoder:** A convolutional autoencoder based on Yu et al. (Yu & Liu, 2020), trained with reconstruction loss on unlabeled data, with the learned bottleneck representations used as features for downstream classification.
- **Manual inspection:** Throughput and accuracy estimates were derived from published industry benchmarks on human visual inspection of wafer maps (Shinde et al., 2022), with an average inspection time of approximately 45 seconds per wafer; the 92.1% accuracy figure reflects mid-range inter-rater agreement reported for trained inspectors.

Complexity Analysis: Our GNN has theoretical complexity $O(|E| \cdot d \cdot k)$ for message passing, where $|E| \approx 4800$ edges for 8-connectivity on 625 nodes, $d = 128$ feature dimension, and $k = 4$ layers. This yields 2.46M FLOPs per forward pass, compared to 4.1B FLOPs for ResNet-50—a 1667× reduction in computational requirements.

Inference Performance: Our 8.3ms inference enables 120 wafers/second, meeting real-time manufacturing requirements. This represents a 5422× speedup over manual inspection while improving accuracy from 92.1% to 98.6%. For high-volume manufacturing lines processing 100,000 wafers/day, this translates to significant throughput improvements without compromising detection quality.

Memory Efficiency: The 387MB footprint enables edge deployment on devices like NVIDIA Jetson Xavier (512MB GPU RAM), without requiring cloud infrastructure. This is 53% smaller than ResNet-50 due to the sparse graph representation. The reduced memory requirement also enables deployment on embedded industrial controllers common in semiconductor fabrication environments.

Scalability Analysis: We evaluated scalability across different wafer sizes and resolutions. For larger wafers (300mm with 50×50 dies), inference time increases to 14.2ms while maintaining 98.4% accuracy. The sparse graph representation scales linearly with die count rather than quadratically as in fully-connected approaches.

Table 7 Statistical Significance of Key Comparisons (paired t-test, 5 runs)

Comparison	Δ Accuracy	p-value	95% CI
Ours vs. Reconstruction Loss	+4.5%	<0.001	[3.8%, 5.2%]
Ours vs. Self-Sup. Rotation	+6.8%	<0.001	[5.9%, 7.7%]
Ours vs. Random Init.	+13.3%	<0.001	[12.1%, 14.5%]
Ours vs. CNN (Nakazawa & Kulkarni, 2018)	+0.4%	0.023	[0.1%, 0.7%]
Comparison	Δ ARI	p-value	95% CI
8-conn. vs. k-NN (k=8)	+0.08	0.002	[0.04, 0.12]
8-conn. vs. Fully-connected	+0.16	<0.001	[0.12, 0.20]

Table 8 Sensitivity to Similarity Thresholds

τ^+ (positive)	τ^- (negative)	ARI	Accuracy
0.5	0.3	0.85	97.1%
0.6	0.2	0.88	98.2%
0.6	**0.3**	**0.89**	**98.6%**
0.6	0.4	0.86	97.8%
0.7	0.3	0.87	98.0%

Bold values indicate statistical significance ($p < 0.05$, paired t-test over 5 random seeds)

Table 9 Sensitivity to Criteria Weights (w_1, w_2, w_3, w_4)

Weights (S_{sp}, S_{den}, S_{str}, S_{pr})	ARI	Accuracy
0.4, 0.3, 0.2, 0.1 (Ours)	**0.89**	**98.6%**
0.25, 0.25, 0.25, 0.25 (Equal)	0.85	97.2%
0.5, 0.2, 0.2, 0.1	0.88	98.3%
0.3, 0.4, 0.2, 0.1	0.87	98.0%
0.4, 0.3, 0.3, 0.0 (No S_{pr})	0.88	98.4%

Bold values indicate statistical significance ($p < 0.05$, paired t-test over 5 random seeds)

Cost Analysis. We provide a transparent, model-based estimate of the operational economics of deploying our system in a representative fab processing 50,000 wafers per month. The figures below are illustrative rather than measured from a specific live deployment; we list every assumption explicitly so that practitioners can re-derive the numbers under their own cost structure (Table 11).

Under these assumptions, deployment yields an approximately 78% reduction in inspection operating cost, with payback well inside one year for a fab of this scale. The 78% headline figure used in the abstract follows directly from this model. We acknowledge that absolute cost figures depend strongly on regional labor rates, capital amortization policy, and existing MES integration cost; the structure of the calculation, rather than the specific dollar values, is the intended contribution of this analysis.

False-positive reduction. The 34% reduction in false-positive rate (mentioned in the abstract) is grounded in studies of human inter-rater reliability for visual wafer inspection, which report disagreement rates in the 5–15% range with substantial inspector-to-inspector variability (Shinde et al., 2022). Our deterministic model produces consistent decisions under identical input, eliminating the inspector-variability component of the false-positive budget. We caution that this is an estimate of the systematic component of FPR reduction; defect-class-specific FPR is reported in the per-class F1 results (Table 5) and the confusion matrix (Figure 6), which are the empirical references for actual model performance.

Discussion

Our experimental results demonstrate that combining GNNs with contrastive learning provides an effective approach for self-supervised wafer defect detection. Several key insights emerge from our analysis:

Effectiveness of Graph Representation: The graph-based representation of wafer maps enables our model to capture spatial relationships that are critical for defect pattern recognition. Unlike CNNs that operate on fixed grid structures, GNNs naturally handle the circular boundary of wafers without introducing padding artifacts. The message passing mechanism allows information to propagate across the wafer, enabling detection of distributed defect patterns that span multiple regions. This is particularly important for patterns like Edge-Ring and Edge-Loc that are defined by their spatial distribution rather than individual die characteristics.

Comparison with CNN Approaches: While CNNs have achieved strong results on wafer defect detection (Nakazawa & Kulkarni, 2018), they treat the wafer map as a regular image, losing the structural information about die connectivity. Our GNN approach explicitly models the topology of the wafer, leading to a 0.4% improvement in accuracy while using 53% less memory than ResNet-50 (387 MB vs. 823 MB). Additionally, the GNN naturally handles the varying sizes and shapes of wafer maps across different manufacturing processes, whereas CNNs require fixed input dimensions with padding or resizing.

Table 10 Computational Comparison of Baseline Methods (evaluated under identical experimental conditions)

Method	Train (h)	Infer (ms)	Mem (MB)	Acc.
Manual (Shinde et al., 2022)	N/A	45000	N/A	92.1%
Traditional CNN (Nakazawa & Kulkarni, 2018)	2.8	5.2	512	95.8%
ResNet-50 (He et al., 2016)	4.1	6.7	823	96.9%
Autoencoder (Yu & Liu, 2020)	3.3	4.8	445	94.1%
GNN (Ours)	**5.2**	**8.3**	**387**	**98.6%**

Bold values indicate statistical significance ($p < 0.05$, paired t-test over 5 random seeds)

Table 11 Cost Model Assumptions and Resulting Operational Economics (per month, 50,000 wafers/month)

Manual inspection (baseline)	
Inspection time per wafer	45 s
Inspector loaded labor rate	US$30/hour
Cost per wafer	US$0.375
Monthly inspection cost	US$18,750
Automated GNN deployment	
Inference time per wafer (RTX 3090)	8.3 ms
GPU server amortized cost (5 yr) + power + maintenance	US$1,000/month
Engineer oversight (∼10% FTE)	US$3,000/month
Total monthly operating cost	US$4,000
Net economics	
Monthly cost reduction	US$14,750 (78.7%)
Initial capital (hardware + integration)	US$50,000
Payback period	3.4 months

Value of Self-Supervised Pre-training: The contrastive learning framework leverages domain-driven heuristics—spatial pattern similarity, defect density proximity, and production context—to construct pseudo-labels for positive and negative pair selection, rather than relying on manual annotations. We acknowledge that these heuristics encode implicit domain knowledge (e.g., defect density captures structural properties correlated with defect types), placing our approach in the self-supervised learning paradigm rather than being purely unsupervised. The subsequent classification stage uses a supervised MLP trained on a fraction of labeled data. Nevertheless, this two-stage design is practically valuable: the self-supervised pre-training stage requires no manual labels, and the ability to maintain 95.7% accuracy with only 25% labeled data demonstrates the practical applicability of our approach, potentially reducing labeling costs by 75%. Moreover, self-supervised pre-training on unlabeled production data enables rapid adaptation to new manufacturing lines without extensive annotation effort.

Industrial Deployment Considerations: Beyond accuracy metrics, practical deployment requires consideration of integration with existing manufacturing execution systems (MES), real-time data pipelines, and operator interfaces. Our framework exports ONNX models compatible with standard inference engines (TensorRT, OpenVINO), enabling integration with industrial control systems. The deterministic inference time (standard deviation <0.3ms) ensures predictable throughput in production environments.

Interpretability: The graph structure provides inherent interpretability through analysis of node importance scores and edge weights. Engineers can examine which regions of the wafer contribute most to the classification decision, facilitating root cause analysis of manufacturing defects. This transparency is crucial for adoption in quality-critical semiconductor manufacturing.

Robustness Analysis: We evaluated robustness to sensor noise by adding Gaussian noise ($\sigma = 0.1$) to input features, observing only 1.8% accuracy degradation.

Out-of-Distribution Detection. We evaluated the model's ability to flag novel defect patterns using a leave-one-class-out (LOCO) protocol: for each of the 9 defect classes in WM-811K, we retrained the MLP classifier on the remaining 8 classes (in-distribution, ID) and treated the held-out class as out-of-distribution (OOD). We use Maximum Softmax Probability (MSP) (Hendrycks & Gimpel, 2017) as the OOD score: an input is flagged as OOD when $\max_c p(c \mid x) < \tau$. The threshold τ is calibrated on the ID validation set so that

Table 12 Out-of-Distribution Detection (Leave-One-Class-Out, MSP@95%TPR$_{ID}$)

Held-out class	Detection rate	AUROC
Center	0.967	0.978
Donut	0.912	0.945
Edge-Ring	0.953	0.971
Edge-Loc	0.881	0.918
Local	0.842	0.891
Near-full	0.876	0.912
Random	0.834	0.879
Scratch	0.901	0.937
None	0.881	0.913
Mean	**0.893**	**0.927**

Bold values indicate statistical significance ($p < 0.05$, paired t-test over 5 random seeds)

95% of ID samples are retained (TPR$_{ID}$ = 0.95), giving a fair operating point for measuring the OOD detection rate.

Table 12 reports per-class OOD detection rates and AUROC. The mean OOD detection rate is 89.3%, with mean AUROC of 0.927. Spatially distinctive patterns (Center, Edge-Ring) are easiest to flag as anomalies when held out (detection rates above 95%), while patterns with high visual similarity to in-distribution classes (Local vs. Edge-Loc, Random vs. Near-full) yield lower rates around 80–85%.

This protocol corresponds to a realistic deployment scenario in which a new defect mode emerges from a process change and the system must flag it for engineering review before retraining. The 10.7% miss rate (1 - 0.893) reflects cases where the held-out class is sufficiently similar to an in-distribution class that the classifier remains confident, and is the limit of MSP-based detection. Improving this further would likely require explicit OOD-aware objectives (e.g., outlier exposure or energy-based scoring) and is identified as future work.

Transferability of the Pipeline. While the empirical evaluation in this paper focuses on WM-811K, the proposed pipeline is constructed from components that are not specific to that dataset. We discuss transferability along three axes: (i) *architectural portability*—the 8-connectivity graph construction and the 9-dimensional node features depend only on the wafer being represented as a regular two-dimensional grid; the GIN+JK encoder is dimension-agnostic and operates on any node-feature dimensionality. No part of the architecture encodes WM-811K-specific assumptions. (ii) *Similarity-criteria portability*—among the four similarity terms in S_{total}, three (spatial pattern, defect density, structural descriptors) are domain-general and can be computed on any wafer-map dataset. The fourth term, production proximity (S_{pr}), depends on the availability of lot identifiers and timestamps, which are recorded automatically by manufacturing execution systems but are absent from public benchmarks such as MixedWM38 (Wang et al., 2020). The weight ablation in Table 9 already shows that setting $S_{pr} = 0$ causes only a marginal drop in clustering quality (ARI 0.89 → 0.88); for datasets without production metadata, S_{pr} can be removed and the remaining weights renormalized with no other change. (iii) *Hyperparameter portability*—the similarity-criteria weights and contrastive margin were tuned on WM-811K and may benefit from per-dataset re-tuning; this is a standard practice rather than a structural limitation.

Empirical Validation on MixedWM38. To complement the discussion above, we additionally evaluated the proposed pipeline on the MixedWM38 dataset (Wang et al., 2020), which contains 38,015 wafer maps spanning 38 classes (1 normal, 8 single-type, and 29 mixed-type defect compositions including 4 four-defect combinations). Two minimal adaptations to the multi-hot label structure of MixedWM38 were required: (i) the categorical classifier was replaced by an 8-bit multi-label head trained with a class-balanced binary cross-entropy loss (one bit per single-type defect, allowing the 29 mixed compositions to be expressed compositionally); and (ii) the production-proximity term S_{pr} was set to zero (lot/timestamp metadata is unavailable for MixedWM38), with the remaining similarity weights renormalized—the configuration whose minor impact was already established by Table 9. All other components—the 8-connectivity graph construction, the GIN+JK encoder, and the contrastive pre-training framework—were reused as-is, and the wafer maps were resized from native 52×52 to 36×36 to remain within the same input regime as the WM-811K experiments. Per-bit isotonic regression (a non-parametric monotone calibrator) was fit on the validation logits to convert raw network outputs into well-calibrated probabilities; this yields the best balance of bit-wise accuracy and F1, and—because the calibrated outputs are already probabilities—the default decision threshold of 0.5 is used.

Table 13 reports the resulting metrics on the held-out 7,603-wafer test set. The pipeline reaches **90.86% bit-wise accuracy** on the 8 defect-type bits with a bit-wise macro-F1 of **0.815**, demonstrating that the underlying representations transfer to the compositional regime of MixedWM38 and reach the operating range of the lower-end published baselines on this dataset (Wang et al. DCNet: 93.20% (Wang et al., 2020)). Bit-wise accuracy is the metric most directly comparable across multi-label methods, since the 38-way categorical decode requires every one of the 8 defect bits to be predicted correctly simultaneously and is therefore much harder; we report both. The substantial gap that remains to the best-performing published methods (WSCN 98.20% (Nag et al., 2022); PIENet 98.90% (Gu et al., 2025)) is attributable to factors not addressed in this experiment: (i) the 36×36 downsampling discards spatial detail that the four-mixed classes

Table 13 Empirical Transferability to MixedWM38 (full data, 30 epochs of self-supervised pre-training + 30 epochs of supervised fine-tuning, 36×36 inputs, 8-bit multi-label head with per-bit isotonic-calibrated probabilities

Metric (test set, $N = 7{,}603$)	Value
Bit-wise accuracy (8-bit per-defect)	**0.9086**
Bit-wise macro-F1	0.8152
Categorical (38-way exact-match) accuracy	0.4260
Categorical macro-F1	0.3473
4-mixed-defect accuracy (hardest 4 classes)	0.2313
4-mixed-defect macro-F1	0.3081
Reference: Wang et al. (Wang et al., 2020) (full pipeline, 4,000 epochs)	0.9320
Reference: WSCN (Nag et al., 2022) encoder-decoder + N-pair contrastive	0.9820

Bold values indicate statistical significance ($p < 0.05$, paired t-test over 5 random seeds)

depend on; (ii) we keep the GIN encoder trained on WM-811K rather than employing per-dataset architecture tuning; and (iii) we do not use the deeper supervised-learning training schedules typical of the published baselines (e.g., 4,000 epochs in DCNet). Closing this gap is identified as concrete follow-up work in Section Conclusion.

Limitations: Despite achieving high accuracy, our model shows reduced performance on: (1) mixed defect patterns combining multiple defect types (3.2% of misclassifications); (2) subtle low-density defects with fewer than 5 defective dies; and (3) novel defect types not represented in training data. The 10.7% missed out-of-distribution detection rate indicates room for improvement in handling completely novel patterns.

Conclusion

This paper presents a novel self-supervised approach combining Graph Neural Networks with contrastive learning for semiconductor wafer defect detection. Our two-stage method—self-supervised pre-training using domain-driven heuristics followed by supervised fine-tuning—achieves 98.6% classification accuracy on the WM-811K dataset, outperforming state-of-the-art supervised methods while substantially reducing reliance on labeled training data.

Key Contributions:

1. First application of contrastive learning with GNNs for wafer defect detection, achieving 98.6% accuracy with self-supervised pre-training that minimizes dependence on labeled data
2. Theoretically-grounded 8-connectivity graph construction that preserves spatial die relationships and handles circular wafer boundaries naturally
3. Multi-criteria similarity scoring combining spatial, density, and pattern-based metrics for domain-appropriate positive pair selection
4. Real-time deployment feasibility with 8.3ms inference time and 387MB memory footprint, enabling edge deployment
5. Demonstrated 78% cost reduction compared to manual inspection with 5422× throughput improvement

Future Directions: Several promising research directions emerge from this work:

- **Multi-modal Integration:** Incorporating process parameters (temperature, pressure, chemical concentrations, equipment identifiers) as additional node features could improve defect root cause identification
- **Temporal Modeling:** Extending to sequences of wafers for detecting gradual equipment degradation and predictive maintenance applications
- **Transfer Learning:** Developing pre-trained models that rapidly adapt to new manufacturing processes with minimal fine-tuning
- **Cross-Dataset Refinement:** An initial transferability study on MixedWM38 (Table 13) reaches 90.86% bit-wise accuracy with only a minor multi-label-head adaptation. Closing the remaining gap to the best-performing published methods on this dataset (98–99%) is a concrete direction for follow-up work, with native 52×52 inputs, attention-based pooling, and longer fully-supervised training schedules identified as the most promising directions
- **Active Learning:** Combining unsupervised clustering with strategic sample selection to minimize labeling requirements while maximizing model improvement
- **Explainability:** Enhancing interpretability through attention mechanisms and GNN explanation methods to support manufacturing engineers in root cause analysis

Our approach demonstrates that combining geometric deep learning with self-supervised objectives provides an effective solution for industrial quality control where labeled data is scarce but domain structure is rich.

Acknowledgements This work was supported by FCT/MECI through national funds and when applicable co-funded EU funds under UID/50008: Instituto de Telecomunicações.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Muhammad Awais. The first draft of the manuscript was written by Muhammad Awais, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by FCT|FCCN (b-on). This work was supported by Fundação para a Ciência e a Tecnologia (FCT/MECI) through national funds and when applicable co-funded by EU funds under UID/50008: Instituto de Telecomunicações.

Data Availibility The WM-811K dataset used in this study is publicly available at https://www.kaggle.com/datasets/qingyi/wm811k-wafermap.

Code availability The complete implementation, including pre-training, fine-tuning, baselines, and scripts to regenerate every table and figure in the paper, is publicly available at https://github.com/irjawais/wafer-gnn-contrastive.

Declarations

Competing interests The authors have no relevant financial or non-financial interests to disclose.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Bai, J., Wu, D., Shelley, T., et al. (2024). A comprehensive survey on machine learning driven material defect detection: Challenges, solutions, and future prospects. https://doi.org/10.48550/arXiv.2406.07880,

Chang, C. Y., Li, C., Chang, J. W., et al. (2009). An unsupervised neural network approach for automatic semiconductor wafer defect inspection. *Expert systems with Applications, 36*(1), 950–958.

Chen, S., Yi, M., Zhang, Y., et al. (2021). A self-adaptive dbscan-based method for wafer bin map defect pattern classification. *Microelectronics Reliability, 123*, Article 114183.

Chen, S., Zhang, Y., Hou, X., et al. (2022). Wafer map failure pattern recognition based on deep convolutional neural network. *Expert Systems with Applications, 209*, Article 118254.

Chen, T., Kornblith, S., & Norouzi, M. (2020). A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning, PMLR* (pp. 1597–1607)

Chou, P. B., Rao, A. R., Sturzenbecker, M. C., et al. (1997). Automatic defect classification for semiconductor manufacturing. *Machine vision and applications, 9*, 201–214.

Edgar, T. F., Butler, S. W., Campbell, W. J., et al. (2000). Automatic control in microelectronics manufacturing: Practices, challenges, and possibilities. *Automatica, 36*(11), 1567–1603.

Ezell, S. (2024). Assessing India's Readiness to Assume a Greater Role in Global Semiconductor Value Chains. Information Technology and Innovation Foundation

Fofanah, A. J., Chen, D., Wen, L., et al. (2024). Addressing imbalance in graph datasets: Introducing gate-gnn with graph ensemble weight attention and transfer learning for enhanced node classification. *Expert Systems with Applications, 255*, Article 124602.

Gu, S., Lei, Z., Zhao, D., et al. (2025). A patch-interactive enhancement network for semiconductor wafer map mixed defect recognition with a two-stage training strategy. *Measurement Science and Technology, 36*(4), Article 045403. https://doi.org/10.1088/1361-6501/adbc0d

He, K., Zhang, X., & Ren, S. (2016). Deep residual learning for image recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 770–778)

He, K., Fan, H., & Wu, Y. (2020). Momentum contrast for unsupervised visual representation learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9729–9738)

Hendrycks, D., & Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. International Conference on Learning Representations (ICLR

Hung, S. C., Banerjee, S., Chaudhuri, A., et al. (2023). Transferable graph neural network-based delay-fault localization for monolithic 3-d ics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 42*(11), 4296–4309.

Hwang, J., & Kim, H. (2020). Variational deep clustering of wafer map patterns. *IEEE Transactions on Semiconductor Manufacturing, 33*(3), 466–475.

Jin, C., Kim, H., Piao, Y., et al. (2020). Wafer map defect pattern classification based on convolutional neural network features and error-correcting output codes. *Journal of Intelligent Manufacturing, 31*(8), 1861–1875.

Kim, J., Lee, Y., & Kim, H. (2018). Detection and clustering of mixed-type defect patterns in wafer bin maps. *IISE Transactions, 50*(2), 99–111.

Kim, T., & Behdinan, K. (2023). Advances in machine learning and deep learning applications towards wafer map defect recognition and classification: a review. *Journal of Intelligent Manufacturing, 34*(8), 3215–3247.

Kong, Y., & Ni, D. (2020). A semi-supervised and incremental modeling framework for wafer map classification. *IEEE Transactions on Semiconductor Manufacturing, 33*(1), 62–71. https://doi.org/10.1109/TSM.2020.2964581

Li, K. S. M., Jiang, X. H., Chen, L. L. Y., et al. (2022). Wafer defect pattern labeling and recognition using semi-supervised learning. *IEEE Transactions on Semiconductor Manufacturing, 35*(2), 291–299. https://doi.org/10.1109/TSM.2022.3159246

Lv, F., Xie, J., & Li, W. (2024). Defect pattern classification of wafer graph based on dual channel convolutional neural network. *2024 6th International Conference on Communications* (pp. 839–842). Information System and Computer Engineering (CISCE): IEEE.

Ma, J., Zhang, T., Yang, C., et al. (2023). Review of wafer surface defect detection methods. *Electronics, 12*(8), 1787. https://doi.org/10.3390/electronics12081787

Ma, M. D., Wong, D. S. H., Jang, S. S., et al. (2009). Fault detection based on statistical multivariate analysis and microarray visualization. *IEEE Transactions on industrial informatics, 6*(1), 18–24.

Nag, S., Makwana, D., Sai Chandra Teja, R., et al. (2022). WaferSeg-ClassNet - a light-weight network for classification and segmentation of semiconductor wafer defects. *Computers in Industry, 142*, Article 103720. https://doi.org/10.1016/j.compind.2022.103720

Nakazawa, T., & Kulkarni, D. (2018). Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing, 31*(2), 309–314.

Oyedotun, O. K., Al Ismaeil, K., & Aouada, D. (2022). Why is everyone training very deep neural network with skip connections? *IEEE Transactions on Neural Networks and Learning Systems, 34*(9), 5961–5975.

Patel, T., Murugan, R., Yenduri, G., et al. (2024). Demystifying defects: Federated learning and explainable ai for semiconductor fault detection. *IEEE Access, 12*, 116987–117007. https://doi.org/10.1109/ACCESS.2024.3425226

Saqlain, M., Jargalsaikhan, B., & Lee, J. (2019). A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 32*(2), 171–182.

Saqlain, M., Abbas, Q., & Lee, J. Y. (2020). A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing, 33*(3), 436–444.

Scarselli, F., Gori, M., Tsoi, A. C., et al. (2008). The graph neural network model. *IEEE transactions on neural networks, 20*(1), 61–80.

Shankar, N., & Zhong, Z. (2003). Rule-based inspection of wafer surface. In: 2003 4th International Conference on Control and Automation Proceedings, IEEE, pp 752–755

Shinde, P. P., Pai, P. P., & Adiga, S. P. (2022). Wafer defect localization and classification using deep learning techniques. *IEEE Access, 10*, 39969–39974.

Shon, H. S., Batbaatar, E., Cho, W. S., et al. (2021). Unsupervised pre-training of imbalanced data for identification of wafer map defect patterns. *IEEE Access, 9*, 52352–52363.

Sun, F. Y., Hoffmann, J., & Verma, V. (2020). Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. International Conference on Learning Representations (ICLR

Wang, J., Xu, C., Yang, Z., et al. (2020). Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition. *IEEE Transactions on Semiconductor Manufacturing, 33*(4), 587–596. https://doi.org/10.1109/TSM.2020.3020985

Wang, L.C., Zeng, Y. (2023). Machine learning support for wafer-level failure pattern analytics. In: Machine Learning Support for Fault Diagnosis of System-on-Chip. Springer International Publishing, p 275–312

Weber, C., Tripuramallu, A., & Czerner, P. (2021). Clustering wafer defect patterns within the semiconductor industry based on wafer maps, using an agile unsupervised deep learning approach. *2021 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1913–1918). IEEE.

Xu, K., Hu, W., & Leskovec, J. (2019). How powerful are graph neural networks? International Conference on Learning Representations

Xuen, L. S., Khairuddin, I. M., & Razman, M. (2022). The classification of wafer defects: a support vector machine with different densenet transfer learning models evaluation. *International conference on robot intelligence technology and applications* (pp. 304–309). Springer International Publishing.

You, Y., Chen, T., & Sui, Y. (2020). Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems (NeurIPS)* (pp. 5812–5823)

Yu, J., & Liu, J. (2020). Two-dimensional principal component analysis-based convolutional autoencoder for wafer map defect detection. *IEEE Transactions on Industrial Electronics,68*(9), 8789–8797.

Zhao, M., Xu, Z., & Ji, J. (2024). Wafer defect recognition method. In: Li T, Nian F, Shan C, et al (eds) Neuroscience-driven Visual Representation. Frontiers Media SA, p 26–39

Zheng, H., Sherazi, S. W. A., Son, S. H., et al. (2021). A deep convolutional neural network-based multi-class image classification for automatic wafer map failure recognition in semiconductor manufacturing. *Applied Sciences, 11*(20), 9769.

Zheng, Y., & Chee, K. W. (2024). Advanced techniques in semiconductor defect detection and classification: Overview of current technologies and future trends in ai/ml integration. *2024 World Rehabilitation Robot Convention (WRRC)* (pp. 1–5). IEEE.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.