Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-025-02775-2

Transductive zero-shot learning for mixed-type defect classification in wafer maps

Jun Liu[1,2] · **Jifei Lu**[1,2] · **Tian Chen**[1,2] · **Xi Wu**[1,2] · **Huaguo Liang**[3] · **Xiaohui Yuan**[4] · **Yen Pham**[4]

Received: 4 October 2025 / Accepted: 12 December 2025

Abstract

In semiconductor manufacturing, rapidly identifying process faults through wafer map defect recognition can significantly improve production yield. However, annotating wafer map defect types—especially complex mixed-type defects—requires skilled technicians and substantial time, leading to high annotation costs. To address this issue, we present a transductive zero-shot learning method that classifies mixed-type defects using only labeled samples of single-type defects. The presented technique eliminates the need for labeled mixed-type defects by leveraging semantic information to bridge known classes (single-type) and unknown classes(mixed-type). This directly translates to reduced time and annotation costs, as well as faster process diagnosis in production environments. We introduce three key strategies to improve classification accuracy: (1) collaborative optimization of the visual feature extractor and semantic embedder, (2) iterative updating of the semantic space, and (3) progressive pseudo-labeling for retraining. Extensive experiments demonstrate that the proposed method substantially surpasses previous transductive zero-shot learning methods, particularly on mixed-type defects.

Keywords Wafer maps · Pattern classification · Mixed-type defect patterns · Transductive zero-shot learning · Pseudo-labeling

✉ Jun Liu
liujun@hfut.edu.cn

Jifei Lu
ljf@mail.hfut.edu.cn

Tian Chen
ct@hfut.edu.cn

Xi Wu
wuxi@hfut.edu.cn

Huaguo Liang
huagulg@hfut.edu.cn

Xiaohui Yuan
xiaohui.yuan@unt.edu

Yen Pham
yenpham@my.unt.edu

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, Anhui, China

[2] Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, Hefei 230601, Anhui, China

[3] School of Microelectronics, Hefei University of Technology, Hefei 230601, Anhui, China

[4] Department of Computer Science and Engineering, University of North Texas, Denton, TX 76207, USA

Introduction

The advancement of Industry 4.0 has catalyzed a paradigm shift towards data-driven intelligence and digital twin technologies in industrial manufacturing (Xie et al., 2022; Feng et al., 2023a). Within this broader context, the semiconductor manufacturing industry presents a critical domain where the rapid and accurate diagnosis of production anomalies is paramount for yield enhancement. Specifically, the automated recognition of defect patterns from wafer maps—a task analogous to vibration-based gear wear monitoring (Feng et al., 2023b) or to surface degradation assessment using digital-twin techniques (Feng et al., 2023a)—serves as a vital tool for pinpointing process faults.

In semiconductor manufacturing, after dies are fabricated on the wafer, a Chip Probing (CP) test is performed to electrically test each die before dicing. Defective dies are marked, generating wafer maps that visually represent their spatial distribution (Jang et al., 2019). Based on these spatial distribution characteristics, wafer map defect patterns are categorized into 9 distinct types. Each defect type represents a different issue that may arise during the manufacturing process (Liu and Chien, 2013). Automating the

Published online: 08 January 2026

classification of these defect patterns enables rapid identification of process anomalies, which is essential for improving production yield. While manual inspection by engineers is both time-consuming and costly, machine learning and deep learning techniques offer a scalable and efficient alternative for automatic defect pattern recognition.

Early methods for wafer map defect pattern classification primarily depended on traditional machine learning techniques with handcrafted features (Kim and Behdinan, 2023). In recent years, deep learning, especially Convolutional Neural Networks (CNNs), has emerged as the predominant approach, as it can automatically extract wafer map features and achieve superior performance (Nakazawa and Kulkarni, 2019). However, as manufacturing processes grow more complex, the mixed-type defects have emerged. Mied-type defect means multiple sigle-type defects are simultaneously present on a wafer (Geng et al., 2023). Figure 1 displays four typical single-type defects (first row) and mixed-type defects (second row).

Compared to single-type defects, mixed-type defect patterns exhibit significantly greater complexity and diversity, posing substantial challenges for accurate classification. The difficulties primarily stem from three factors. First, mixed-type defects often display high inter-class similarity and significant intra-class diversity due to differences in the composition ratio and spatial distribution of the defects (Chen et al., 2024). Second, the morphological characteristics of mixed-type defects are highly stochastic, varying in location, size, and rotation angle. Furthermore, patterns from different types of defects may overlap and interfere with one another. This complexity hinders Convolutional Neural Networks from effectively disentangling and extracting the essential features of each defect type, leading to a notable decline in model discriminability, particularly when dealing with complex patterns involving three or four mixed defects (Luo and Wang, 2023).

Although supervised learning shows promising performance in classifying mixed-type defect wafer maps, its effectiveness heavily depends on large quantities of labeled samples, which pose a significant challenge due to the high cost of annotation (Shim et al., 2021). Labeling single-type defects is comparatively straightforward, as their defect patterns are simple and well defined. By contrast, annotating mixed-type defect wafer maps—where a sample contains multiple defect classes—is substantially more difficult and time-consuming due to complex and often ambiguous morphologies (Wang et al., 2020). If open-set recognition techniques (Liu et al., 2020; Yue et al., 2021) are used to classify mixed-type defects—so that a model can be trained without using mixed-type defect samples while still achieving effective classification accuracy on those defects—the labeling costs associated with mixed-type defects could be substantially reduced.

Several studies have explored open-set recognition techniques to detect the presence of unseen defect types in wafer maps (Jang et al., 2020; Jang and Lee, 2023). However, these methods are inherently limited to identifying the presence of novel defects, rather than providing specific classifications. Baek et al. (2025) introduced a technique based on a contrastive loss function to classify previously unseen defects. However, their method can classify only unseen single-type defects and cannot handle unseen mixed-type defects.

Although prior work has advanced wafer map defect patterns classification, a clear research gap remains in effectively handling mixed-type defects. Current deep-learning models perform well on single-type defects but degrade substantially when encountering mixed-type defects. Moreover, the methods that improve mixed-type defects classification typically demand substantial volumes of labeled samples, and the cost of annotating mixed-type defects is prohibitively high. Notably, the techniques that can classify mixed-type defects without relying on such defect samples remain predominantly uninvestigated.

In this paper, we present a Transductive Zero-Shot Learning-based Wafer Map Classification (ZSWMC) method. The ZSWMC method can classify mixed-type defects only using labeled single-type defect samples and unlabeled

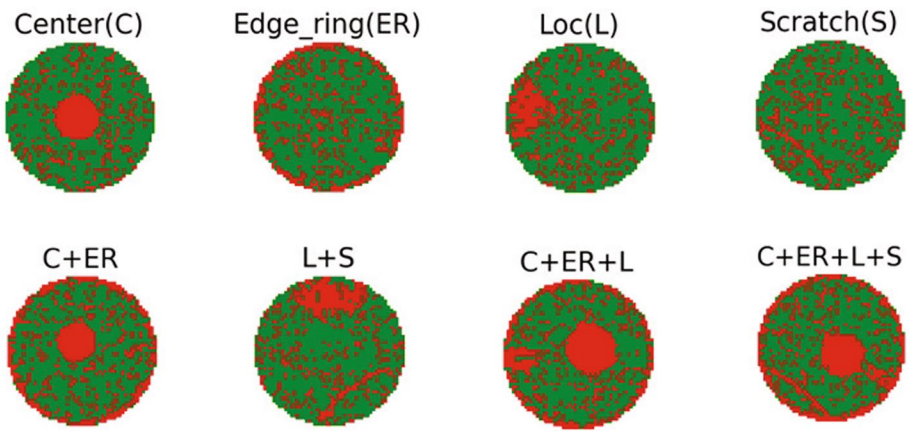


Fig. 1 Several examples of defect patterns

mixed-type defect samples during training. To tackle the identified challenges, we introduce three key optimization strategies: (1) collaborative optimization of the visual feature extractor and semantic embedder, (2) iterative updating of the semantic space, and (3) progressive pseudo-labeling for model retraining. These strategies are designed to enhance semantic-visual alignment and improve classification accuracy for complex mixed-type defects in a real-world setting.

The contributions of the proposed ZSWMC method are summarized as follows:

1. A transductive zero-shot learning approach is presented, enabling the classification of mixed-type defects without requiring labeled mixed-type defect wafer maps during training.
2. The visual feature extractor and semantic embedder are integrated into an end-to-end model for collaborative optimization, leveraging the close semantic correlation between single-type defects and mixed-type defects. Furthermore, the updating strategy of semantic space during the training process is proposed to improve classification accuracy.
3. A progressive pseudo-labeling for model retraining strategy is employed, utilizing the progressive relationships among mixed-type defects to further enhance classification accuracy.

The structure of this paper is as follows. Section "Related work" reviews related research on wafer map defect patterns recognition. Section "The ZSWMC method" presents the proposed ZSWMC methodology. Sections "Experimental setup" and "Analysis of experimental results" describe the experimental setup and discuss the corresponding results, respectively. Concluding remarks are offered in "Conclusion" section.

Related work

Evolution of wafer map defect patterns recognition

Initial methods for recognizing defects in wafer maps depended mainly on conventional machine learning techniques that utilized manually engineered features. For example, (Wu et al., 2015) combined Radon transformation features with geometric attributes, employing a Support Vector Machine (SVM) for classification. Similarly, (Cheng et al., 2022) utilized defect density, yield metrics, and Hough transform features for rapid pattern identification using decision trees. However, these methods were limited by their dependence on manually designed features, which often failed to capture high-level semantic information present in complex defect patterns.

The rapid progress in deep learning has led to its growing application in wafer map recognition. Nakazawa and Kulkarni (2018) were the first to apply Convolutional Neural Networks (CNNs) to this task. Subsequent studies focused on improving feature representation: Chen et al. (2022) integrated Deep Convolutional Neural Networks (DCNNs) with decision-level entropy, while a later work Chen et al. (2023) incorporated an improved convolutional block attention module to prioritize salient defect regions. Additionally, Yi et al. (2024) combined deep features with hand-made density and Radon features, achieving higher accuracy using modified extreme learning machines. To address the common issue of sample imbalance, To address the common issue of imbalanced defect samples, Hyun and Kim (2020) proposed a memory-enhanced CNN architecture with a triplet loss function. Although these supervised methods achieve high accuracy on single-type defects, they exhibit two inherent limitations. First, their performance is highly dependent on large volumes of accurately annotated samples, the acquisition of which is often prohibitively expensive. Second, because these methods are primarily designed for single-type defects, their classification accuracy deteriorates substantially when applied to mixed-type defects.

To tackle the classification of mixed-type defects, dedicated supervised models have been developed. Wang et al. (2020) introduced a deformable convolution network to selectively focus on defect regions, combined with a multi-label output layer that decomposes mixed-type defects into constituent single-type defects using a one-hot encoding mechanism. To tackle the confusion between certain defect types within mixed defects, (Luo and Wang, 2023) proposed the CWDR-Net, which leverages multi-view dynamic feature enhancement and attention mechanisms to improve the classification accuracy of complex mixed-type defects. Although these studies have improved the classification accuracy of mixed-type defects, they still rely heavily on annotated samples. The annotation cost for mixed-type defects is substantially higher than that for single-type defects.

To alleviate the data annotation bottleneck, weakly-supervised paradigms like self-supervised and few-shot learning have been presented. In self-supervised learning, Liao et al. (2022) incorporated self-supervised reconstruction training into the classification task, which effectively utilized pixel-level information from wafer maps, greatly enhancing classification accuracy. The Wafer Map Deep Clustering (WMDC) model, proposed by Xu et al. (2024), learns general representations through unsupervised pre-training. Using a prototype metric loss, it extracts semantic features of defect categories, effectively enhancing recognition

accuracy when transferred to tasks with scarce labeled data. Kahng and Kim (2021) introduced a general self-supervised training framework utilizing noise-contrastive estimation, demonstrating its effectiveness in improving model performance under label scarcity. Both Kim and Kang (2021) and Kwak et al. (2023) adopted contrastive learning strategies. Kim and Kang (2021) employed a Dirichlet Process Mixture model to dynamically generate pseudo-labels for some samples used in subsequent self-supervised training. Kwak et al. (2023) designed a new loss function that integrates label information from the pre-training phase, thereby learning representations that are better suited to classification tasks. To handle mixed-type defects, Wang et al. (2024) introduced a masked autoencoder-based framework, enabling few-shot classification of complex mixed-type defects.

In the realm of few-shot learning, Geng et al. (2021) integrated few-shot learning with self-supervised learning to tackle data imbalance and improve utilization of unlabeled data. However, their method is still limited to single-type defect classification. Liang et al. (2024) proposed a few-shot learning approach for mixed-type defects, employing a masked autoencoder for unsupervised feature learning and a dynamic multi-loss mechanism to enlarge inter-class differences and minimize intra-class variation, thereby improving mixed-type defect classification accuracy. However, these weakly-supervised methods primarily focus on learning features from data and cannot classify previously unseen defects during training.

Detection of unseen defect types

The increasing complexity of chip manufacturing necessitates the identification of novel and unseen defect types. Zhao et al. (2024) proposed an incremental learning-based online detection method, PIRB, which is capable of detecting unseen defects. Alawieh et al. (2020) proposed a deep selective learning model incorporating a rejection mechanism, enabling the model to refrain from making classifications under conditions of high uncertainty. Although these methods effectively signal the presence of a novel defect, they cannot identify its type of defect. The method proposed in Baek et al. (2025) introduced a contrastive loss function to classify previously unseen defects, which can recognize a novel single-type defect. However, it is limited to single-type defect scenarios and cannot classify mixed-type defects.

Many studies employ semantic segmentation to classify mixed-type defects. Yan et al. (2023) employed semantic segmentation to decompose mixed-type defects into their constituent single-type defect components. Chiu and Chen (2021) and Kim et al. (2022) classify mixed-type defects by using labeled single-type defects as training samples. Chiu and Chen (2021) integrated mask R-CNN with rotation data augmentation to enhance classification accuracy on single-type defects, enabling precise classification and localization of defects within mixed-type wafer maps through transfer learning. Kim et al. (2022) proposed a novel single-stage detector that directly learns features from single-type defects and generalizes to mixed-type defects detection via multi-scale feature transfers. Although these semantic segmentation methods do not require labeled mixed-type defect samples, they still depend on extensive pixel-level annotations of single-type defects, necessitating precise defect boundary labeling, which also results in high annotation costs.

Zero-shot learning for defect recognition

Zero-Shot Learning (ZSL) offers a promising alternative by eliminating the need for pixel-level annotations of unseen classes. The principle of ZSL involves utilizing the high-level semantic information, such as attribute or class descriptor vectors, to facilitate the identification of unseen classes during the training (Lampert et al., 2009). Mainstream ZSL methods learn an embedding function that maps both visual features and semantic descriptors into a shared latent space, where classification is performed via similarity metrics.

Kim and Shim (2024) pioneered the application of ZSL to classify unseen single-type defects in wafer maps. However, their method does not address the classification of unseen mixed-type defects. Our proposed method aims to solve this problem through ZSL.

In ZSL, the major challenge is the domain shift problem (Wan et al., 2021), where distributional discrepancies between seen and unseen classes degrade model performance. To mitigate the issue of domain shift, researchers have proposed transductive zero-shot learning methods. These methods leverage labeled samples from seen classes and unlabeled samples from unseen classes for model training, optimizing the embedding function, and improving the model's capacity for generalization. Some representative transductive zero-shot methods, such as VSC (Wan et al., 2021), QFSL (Song et al., 2018), Bi-VAEGAN (Wang et al., 2023), AD3C-FGN (Zhang et al., 2023), and TFVAE-GAN (Narayan et al., 2020), have shown effectiveness in improving classification accuracy for unseen classes.

Our work extends zero-shot learning to the classification of mixed-type defects by treating labeled single-type defects as seen classes and unlabeled mixed-type defects as unseen classes, significantly reducing the annotation cost for mixed-type defects. However, conventional transductive zero-shot learning methods suffer from suboptimal performance due to the separate optimization of visual feature extraction and semantic embedding, which leads

to significant differences between the visual spaces and semantic spaces. To overcome this limitation, the proposed ZSWMC method leverages the close semantic relationships between single-type and mixed-type defects, integrating visual feature extraction and semantic embedding for co-optimization, and iteratively updating the semantic space to bridge the gap. The close semantic relationships mean that mixed-type defects are combinations of single-type defects. For example, a scratch defect typically exhibits a thin-line shape, whereas a center defect is characterized by a solid-circle appearance. The mixed-type defect `C+S` (`Center + Scratch`) displays both thin-line and solid-circle attributes in the wafer maps. Furthermore, the proposed method employs a progressive pseudo-labeling and retraining strategy to fully leverage unlabeled samples, further enhancing classification accuracy.

The ZSWMC method

The ZSWMC method first requires the definition of the semantic center vectors for all single-type and mixed-type defects based on expert knowledge. These vectors function as training targets for the model and serve as the basis for distinguishing between different defects. When classifying wafer map defect categories, the classification model extracts visual features from wafer maps and embeds them into the semantic space to generate semantic vectors. The defect category of each wafer map is determined via calculating the cosine similarity between its semantic vector and the predefined semantic center vectors. The workflow of the ZSWMC method is shown in Fig. 2, comprising four steps.

The first step involves denoising wafer maps from seen classes to preserve the primary defect features. Initially, the seen class contains only samples of single-type defects. The ZSWMC method uses a window threshold (Wang and Chen, 2019) to reduce the noise in wafer maps. The denoising procedure is described in Sect. "Dataset".

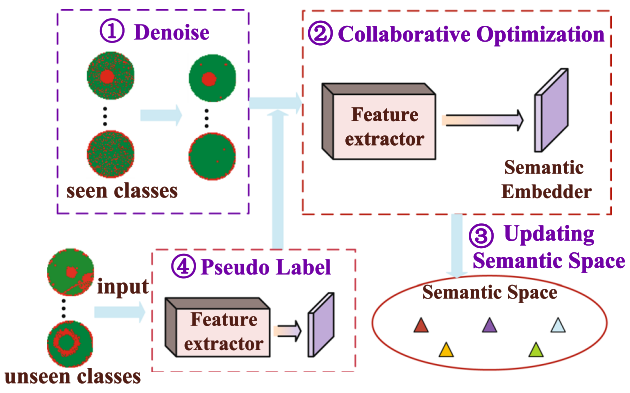


Fig. 2 The workflow of the ZSWMC Method

The second step employs an end-to-end collaborative optimization strategy to jointly optimize the feature extractor and semantic embedder. The feature extractor extracts visual features from input samples, while the semantic embedder maps these visual features into semantic space to generate semantic vectors.

The third step updates the semantic space by recalculating all semantic center vectors to minimize the mapping bias of the visual feature vectors. In Fig. 2, colored triangles (▲) in the semantic space represent the semantic center vectors of different defect categories.

The fourth step implements a progressive pseudo-labeling and retraining strategy. During this step, mixed-type defect wafer maps are assigned pseudo-labels, and a subset of high-confidence pseudo-labeled samples is selected and combined with seen class samples to retrain the model. The specific operations involved in each step are elaborated upon in the subsequent sections.

End-to-end model collaborative optimization

This step utilizes the close semantic correlation between single-type and mixed-type defects to integrate the visual feature extractor and semantic embedder into an end-to-end model for collaborative optimization.

Problem formulation

The matrix $x \in \{0,1,2\}^{m\times n}$ represents a wafer map sample, where 0 denotes background, 1 signifies a good die, and 2 indicates a bad die. The set of seen classes $D_s = \{(x_i, y_i, c_{y_i})\}_{i=1}^{N}$ consists of labeled single-type defect wafer maps, where x_i denotes the i-th wafer map, y_i specifies its defect category represented as an integer, c_{y_i} is the semantic center vector of the defect category y_i, and N indicates the quantity of seen class samples. The unseen class set $D_u = \{x_j\}_{j=1}^{M}$ contains all unlabeled mixed-type defect wafer maps, where x_j represents the j-th wafer map, and M denotes the quantity of unseen class samples.

We establish a mapping from wafer maps to the semantic attribute space. Let $f(\cdot;\theta_f)$ denote the visual feature extractor parameterized by θ_f, and $g(\cdot;\theta_g)$ denote the semantic embedder parameterized by θ_g. Given an input wafer map x_i from a seen class, its semantic vector $\hat{\mathrm{a}}_i$ is obtained by:

$$\hat{\mathrm{a}}_i = g(f(x_i;\theta_f);\theta_g) \tag{1}$$

The training objective is to jointly learn the parameters $\{\theta_f, \theta_g\}$ such that the semantic vector $\hat{\mathrm{a}}_i$ closely approximates its corresponding semantic center vector. Let y_i be the defect type of wafer map x_i, and let c_{y_i} be the semantic

center vector for defect type y_i. This objective is achieved by minimizing the Mean Squared Error (MSE) loss between $\hat{a}_i$ and c_{y_i} across a batch of N samples:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{a}_i - c_{y_i}\|_2^2 \tag{2}$$

where $\|\cdot\|_2$ denotes the $L2$-norm. To mitigate overfitting, the loss function $\mathcal{L}$ comprises the MSE loss and an $L2$ regularization term:

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \underbrace{\lambda\left(\|\theta_f\|_2^2 + \|\theta_g\|_2^2\right)}_{\text{L2 Regularization}} \tag{3}$$

By minimizing $\mathcal{L}$, the model encourages the semantic vectors to align with their corresponding semantic center vectors while constraining the complexity of θ_f and θ_g to improve generalization.

During the test phase, the model first generates the semantic vector of the sample x_t: $\hat{a}_t = g(f(x_t))$. It then computes the cosine similarity between $\hat{a}_t$ and all semantic center vectors. The semantic center vector exhibiting the highest similarity is selected, and its corresponding defect type is assigned to x_t:

$$y_t = \arg\max_{1\le k\le K} \frac{\hat{a}_t \cdot c_k}{\|\hat{a}_t\|_2 \|c_k\|_2} \tag{4}$$

In Eq. 4, K represents the number of defect categories, including seen and unseen classes. The y_t is the predicted defect type of the sample x_t, the symbol denotes the dot product, and c_k represents the semantic center vector of the defect type k.

Collaborative optimization architecture

The architecture for collaborative optimization is illustrated in Fig. 3. We employ the convolutional layers of *ResNet-18* (He et al., 2016) as the visual feature extractor $f(\cdot;\theta_f)$ and a single fully connected(FC) layer as the semantic embedder $g(\cdot;\theta_g)$. During the preliminary training stage, we leverage the set of seen classes $\mathcal{D}_s$, which contains only single-type defect wafer maps with ground-truth labels, to jointly learn the parameters θ_f and θ_g.

The reason that we use a single FC layer as the semantic embedder g is threefold. First, the *ResNet-18* backbone produces a 512-dimensional feature vector from its penultimate layer, providing a high-level visual representation that aligns with the abstraction level of the target semantic attributes. A single linear transformation is therefore sufficient to map between these two high-level spaces. Second, the single FC layer, which contains only $512 \times 20 + 20 = 10,260$ parameters, significantly reduces the model complexity and mitigates the risk of overfitting compared to a deeper network. Finally, since the target semantic space is low-dimensional (20 dimensions) and does not require a highly non-linear mapping, a single FC layer has been widely adopted and proven effective in analogous attribute learning tasks (Xu et al., 2022; Li et al., 2023; Xian et al., 2018).

A standard approach in zero-shot learning involves a two-stage training procedure, where the visual feature extractor and semantic embedder are optimized in sequence. This approach can be formalized using the following two independent objectives:

$$\text{Stage 1:} \quad \min_{\theta_f} \mathcal{L}_{\text{cls}}\left(f(x_i;\theta_f), y_i\right) \tag{5}$$

$$\text{Stage 2:} \quad \min_{\theta_g} \mathcal{L}_{\text{embed}}\left(g(f(x_i;\hat{\theta}_f);\theta_g), c_{y_i}\right) \tag{6}$$

Here, $\mathcal{L}_{\text{cls}}$ is a standard classification loss (e.g., cross-entropy) used to pre-train the visual feature extractor f, yielding parameters $\hat{\theta}_f$. Subsequently, the loss function $\mathcal{L}_{\text{embed}}$ (e.g., MSE) is applied to train the semantic embedder g, optimizing θ_g while keeping $\hat{\theta}_f$ fixed. This training paradigm arises from the semantic discrepancy between seen and unseen classes. The underlying rationale is that jointly optimizing θ_f and θ_g could lead the learned feature space to overfit the seen classes, consequently compromising its generalization capability to unseen classes.

In wafer map defect classification, mixed-type defects are formed by combining multiple single-type defects. This inherent relationship allows their visual and semantic features to be derived from the features of their constituent

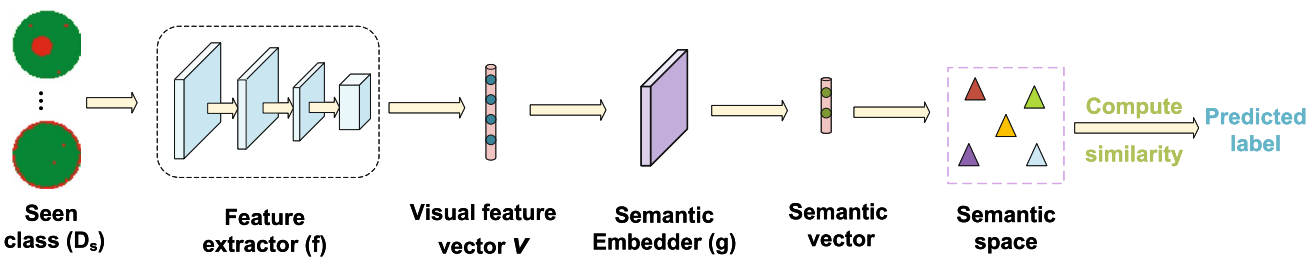


Fig. 3 Collaborative optimization for feature extractor and semantic embedder

single-type components. Therefore, we collaboratively optimize the parameters θ_f and θ_g by minimizing a unified objective function:

$$\min_{\theta_f,\theta_g} \frac{1}{N}\sum_{i=1}^{N} \|g(f(x_i;\theta_f);\theta_g) - \mathrm{c}_{y_i}\|_2^2 + \lambda\left(\|\theta_f\|_2^2 + \|\theta_g\|_2^2\right) \tag{7}$$

This collaborative optimization strategy encourages the visual feature extractor f to learn representations that are inherently aligned with the semantic attribute space. Consequently, when the model is transferred to classify unseen mixed-type defects, both the visual feature extractor and semantic embedder are already tuned to the constituent components of the mixed-type defects. This enables more accurate projection of mixed-type defect patterns into the semantic space, ultimately enhancing classification performance.

Update semantic space

In Sect. "End-to-end model collaborative optimization", both f and g are trained on sample of the seen class (single-type defect wafer maps) by Eq. 7. The classification accuracy is high for seen class samples but significantly decreases when classifying unseen class samples (mixed-type defect wafer maps). To improve classification precision, we propose the semantic space updating strategy comprising three steps, as illustrated in Fig. 4.

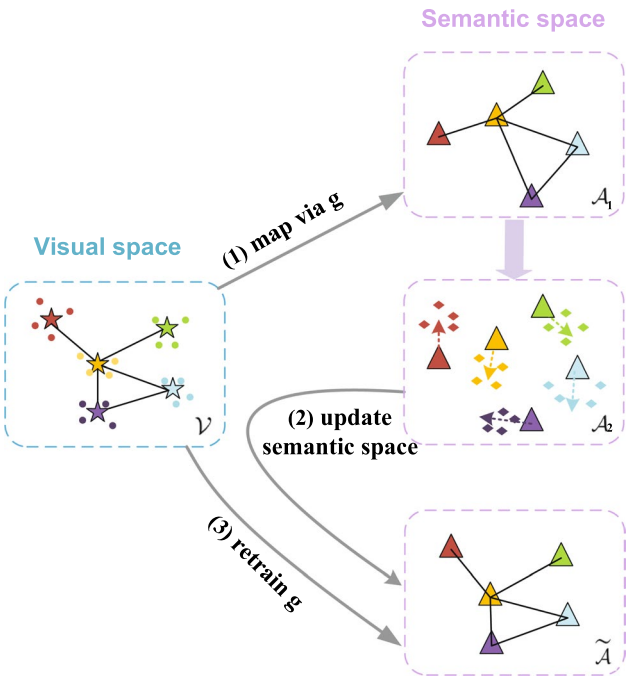


Fig. 4 The schematic diagram of updating semantic space

Map visual feature vectors

The first step maps all visual feature vectors in visual space $\mathcal{V}$ into semantic space $\mathcal{A}_1$, generating the semantic vectors via the semantic embedder g. Formally, for a visual feature vector $\mathrm{v}_i \in \mathcal{V}$ extracted by f, its corresponding semantic vector a_i in semantic space $\mathcal{A}_1$ is obtained by $\mathrm{a}_i = g(\mathrm{v}_i)$.

As shown in Fig. 4, the visual space $\mathcal{V}$ contains visual feature vectors of wafer map samples, extracted by the visual feature extractor f. In this space, samples from different defect categories are represented by colored circles (●), while the colored stars (★) denote the visual center of each category, computed as the mean of all its sample vectors. In the semantic space $\mathcal{A}_1$, colored triangles (▲) represent the semantic center vectors of different defect categories. Initially, the semantic space $\mathcal{A}_1$ contains only these semantic center vectors. After semantic vectors are generated, a new semantic space $\mathcal{A}_2$ is constructed, where colored diamonds (♦) represent semantic vectors corresponding to different defect categories.

Update the semantic center vectors

The second step involves iteratively updating the semantic space, where the semantic center vectors within $\mathcal{A}_2$ are updated. We adopt the core concept of manifold learning (Tenenbaum et al., 2000; Roweis and Saul, 2000; Belkin and Niyogi, 2003): when the semantic vectors of certain samples share a local manifold structure with a semantic center vector, those samples are likely to belong to the same defect type. The principle for updating semantic center vectors is illustrated in the $\mathcal{A}_2$ of Fig. 4. The arrows (→) on the triangles (▲) indicate the update direction, moving the semantic center vectors toward the centroid of their nearest m semantic vectors. Specifically, each semantic center vector is updated to be the average of its nearest m semantic vectors.

For each semantic center vector $\mathrm{c}_k^{(t)}$ at iteration t, we identify its m nearest semantic vectors from the set $\{\mathrm{a}_i\}_{i=1}^{N}$. Let $\mathcal{N}_m(\mathrm{c}_k^{(t)})$ denote the indices of these m nearest neighbors:

$$\mathcal{N}_m(\mathrm{c}_k^{(t)}) = \{i_1, i_2, \ldots, i_m \mid \mathrm{a}_i \in m\text{-NN of } \mathrm{c}_k^{(t)}\} \tag{8}$$

The semantic center vector $\mathrm{c}_k^{(t)}$ is then updated as the mean of its m nearest semantic vectors:

$$\mathrm{c}_k^{(t+1)} = \frac{1}{m} \sum_{j \in \mathcal{N}_m(\mathrm{c}_k^{(t)})} \mathrm{a}_j \tag{9}$$

This update process moves each semantic center vector toward the dense region of its corresponding manifold, making the semantic vector closer to its semantic center vector. After the update, the semantic vectors in $\mathcal{A}_2$ are removed, generating a new semantic space $\widetilde{\mathcal{A}}^{(t+1)} = \{\mathrm{c}_1^{(t+1)}, \mathrm{c}_2^{(t+1)}, \ldots, \mathrm{c}_K^{(t+1)}\}$ containing only updated semantic center vectors.

Retrain semantic embedder

During the collaborative optimization step, we first obtain an initial semantic embedder g that maps visual feature vectors to corresponding semantic vectors. In the third step, the semantic embedder must be retrained to accommodate the updated semantic center vectors. In the retraining process, the parameters of the visual feature extractor f are kept fixed, while the semantic embedder is optimized according to the objective function:

$$\min_{\theta_g} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathrm{c}_{y_i}^{(t+1)} - g(\mathrm{v}_i; \theta_g) \right\|_2^2 + \lambda \|\theta_g\|_2^2 \tag{10}$$

where $\mathrm{c}_{y_i}^{(t+1)}$ is the updated semantic center vector corresponding to defect type y_i of sample i, and λ denotes the regularization coefficient. The first term in Eq. 10 reduces the discrepancy between the semantic vectors and the refined semantic centroids, whereas the second term mitigates the overfitting by regularizing parameters θ_g.

Steps 1 through 3 are repeated until the semantic embedder g converges or the predefined maximum iteration count T is achieved. This iterative process progressively refines the semantic space to better align with the data manifold, thereby improving the classification accuracy for both the seen and unseen classes.

The complete process for the semantic space updating is shown in Algorithm 1. Lines 3 to 7 perform the update of semantic center vectors, and line 9 optimizes the semantic embedder according to Eq. 10. The algorithm ends when $\widetilde{g}$ converges or when the T is achieved. At this point, the final $\widetilde{g}$ and $\widetilde{\mathcal{A}}$ are designated as the semantic embedder and semantic space, respectively.

Require: $\mathcal{V}$, $\mathcal{A}$, g
Ensure: Optimized $\widetilde{g}$, updated $\widetilde{\mathcal{A}}$
1: Initialize $\widetilde{\mathcal{A}} \leftarrow \mathcal{A}$, $\widetilde{g} \leftarrow g$
2: **for** iteration $t = 1$ to T **do**
3: **for** each semantic center vector $c_k \in \widetilde{\mathcal{A}}$ **do**
4: Find its m nearest neighbors in $\widetilde{g}(\mathcal{V})$
5: Compute mean: $\widetilde{c}_k \leftarrow \frac{1}{m} \sum_{j=1}^{m} \text{neighbor}_j$
6: Update $c_k \leftarrow \widetilde{c}_k$
7: **end for**
8: Update $\widetilde{\mathcal{A}} \leftarrow \{c_k \mid \forall k\}$
9: Learn $\widetilde{g}$ by Eq. 10
10: **if** converged **then**
11: **break**
12: **end if**
13: **end for**
14: **return** $\widetilde{g}$, $\widetilde{\mathcal{A}}$

Algorithm 1 Iterative Optimization

Progressive pseudo-labeling and retraining

In mixed-type defects, two-mixed defects are coupled from single-type defects, while three-mixed defects are formed from two-mixed defects by incorporating an additional single-type defect. Therefore, there is a progressive relationship among the defects. After training the model with labeled single-type defects, we observed that it achieved the highest accuracy in recognizing two-mixed defects. Hence, we propose a progressive pseudo-labeling and retraining strategy, as detailed in Algorithm 2.

Require: D_s, D_u, model M, threshold τ
Ensure: Final model M^*
1: $M^* \leftarrow M$
2: **for** $k = 2$ to 4 **do**
3: $C_k \leftarrow \{x \in D_u \mid M^*(x) \text{ predicts } k\text{-mixed}\}$
4: $P_k \leftarrow \{x \in C_k \mid \text{Confidence}(M^*(x)) > \tau\}$
5: Assign pseudo-labels to P_k
6: $D_u \leftarrow D_u - P_k$
7: $D_s \leftarrow D_s \cup P_k$
8: $M^* \leftarrow \text{Train}(M^*, D_s)$
9: **end for**
10: **return** M^*

Algorithm 2 Pseudo-Labeling

First, the model classifies all mixed-type defect wafer maps in D_u, selects the wafer maps with pseudo-labels indicating two-mixed defects, and places them into the set C_2 (Line 3). The samples in C_2 with classification confidence exceeding threshold τ are then filtered into the set P_2 (Line 4). These samples with pseudo-labels in P_2 are transferred from D_u to D_s (Lines 5–7). Subsequently, the procedures described in Sects. "End-to-end model collaborative optimization" and "Update semantic space" are repeated to retrain the model (Line 8), sequentially assigning pseudo-labels to three-mixed and four-mixed defect wafer maps. During this progressive pseudo-labeling process, a high-confidence threshold is used to ensure the quality of the pseudo-labels, gradually optimizing the model's classification ability for mixed-type defects.

Table 1 Attribute information of wafer maps

att1: Solid_circle	att11: Over_90%_defective
att2: Dense	att12: Random_patterns
att3: Cluster_without_center	att13: Chemical_mechanical_polishing
att4: Localized_cluster	att14: Particles
att5: Thin_line_shape	att15: Human errors
att6: Symmetric_to_rotation	att16: Deposition
att7: located_near_an_edge	att17: Rapid_thermal_annealing
att8: Annular_shaped	att18: Photo_lithography
att9: Multiple_appearance	att19: Lay-wise_misalignment
att10: No_defect	att20: Uneven cleaning

Experimental setup

Dataset

The experiments are conducted on the MixedWM38 dataset (Wang et al., 2020), which contains 38 defect categories: 9 single-type defects, 13 two-mixed defects, 12 three-mixed defects, and 4 four-mixed defects. Most defect categories contain 1000 samples, except for `Nearfull` defect samples (866), Random samples (149), and `C+EL+S` samples (2000). In the experiments, the `C+EL+S` samples are downsampled to 1000, while the sample counts of all other categories remain unchanged. During model training, only single-type defect wafer maps have ground truth labels. The dataset is partitioned into training, validation, and test subsets with a ratio of 70, 15, and 15%, respectively. The 9 single-type defects were designated as labeled seen classes, while the 29 mixed-type defects were treated as unlabeled unseen classes.

We utilize a window threshold (Wang and Chen, 2019) to reduce noise in the wafer maps. For each wafer map, a 3×3 sliding window is centered on each defective die to calculate the proportion of defective dies within the window. If this proportion falls below a specified threshold value, the central die is reclassified as defect-free. In contrast, when the proportion exceeds the threshold, the central die remains defective. In the ZSWMC experiments, the threshold value is set to 4/9. Furthermore, all wafer maps are resized to a dimension of 224×224 pixels to conform to the input specifications of the ResNet-18 architecture.

Table 2 Defect types and their semantic center vectors

Defect type	Name	att1	att2	att3	att4	att5	att6	att7	att8	att9	att10	att11	att12	att13	att14	att15	att16	att17	att18	att19	att20
Single-type defects	Center	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0
	Donut	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	Edge_loc	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	1	0	0	1
	Loc	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1
	Edge_ring	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0
	Scratch	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0
	Random	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	Nearfull	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	Normal	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Two-mixed defects	C+EL	1	0	0	1	1	1	1	0	0	0	0	0	1	1	0	1	1	0	0	1
	C+ER	1	0	0	1	1	1	1	1	0	0	0	0	1	0	0	1	1	1	1	0
	C+L	1	0	1	1	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1
	C+S	1	0	0	1	1	1	0	0	1	0	0	0	1	1	1	1	0	0	0	0
	D+EL	0	1	0	0	1	1	1	1	0	0	0	0	0	1	0	1	1	1	0	1
	D+ER	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0
	D+L	0	1	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0	1
	D+S	0	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0
	EL+L	0	0	1	1	1	0	1	0	1	0	0	0	0	1	0	1	1	0	0	1
	EL+S	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	1	0	0	1
	ER+L	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0
	ER+S	0	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1	1	1	0
	L+S	0	0	1	1	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1
Three-mixed defects	C+EL+L	1	0	1	1	1	1	1	0	1	0	0	0	1	1	0	1	1	0	0	1
	C+EL+S	1	0	0	1	1	1	1	0	1	0	0	0	1	1	1	1	1	0	0	1
	C+ER+L	1	0	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	0
	C+ER+S	1	0	0	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	0
	C+L+S	1	0	1	1	1	1	0	0	1	0	0	0	1	1	1	1	0	0	0	1
	D+EL+L	0	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	1	1	0	1
	D+EL+S	0	1	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	1
	D+ER+L	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0
	D+ER+S	0	1	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1	1	1	0
	D+L+S	0	1	1	1	1	1	0	0	1	0	0	0	1	1	1	1	0	0	0	1
	EL+L+S	0	0	1	1	1	0	1	0	1	0	0	0	1	1	1	1	1	0	0	1
	ER+L+S	0	0	1	1	1	1	1	1	1	0	0	0	1	0	0	0	1	1	1	0
Four-mixed defects	C+EL+L+S	1	0	1	1	1	1	1	0	1	0	0	0	1	1	1	1	1	0	0	1
	C+ER+L+S	1	0	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	0
	D+EL+L+S	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	1
	D+ER+L+S	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	1	1	1	0

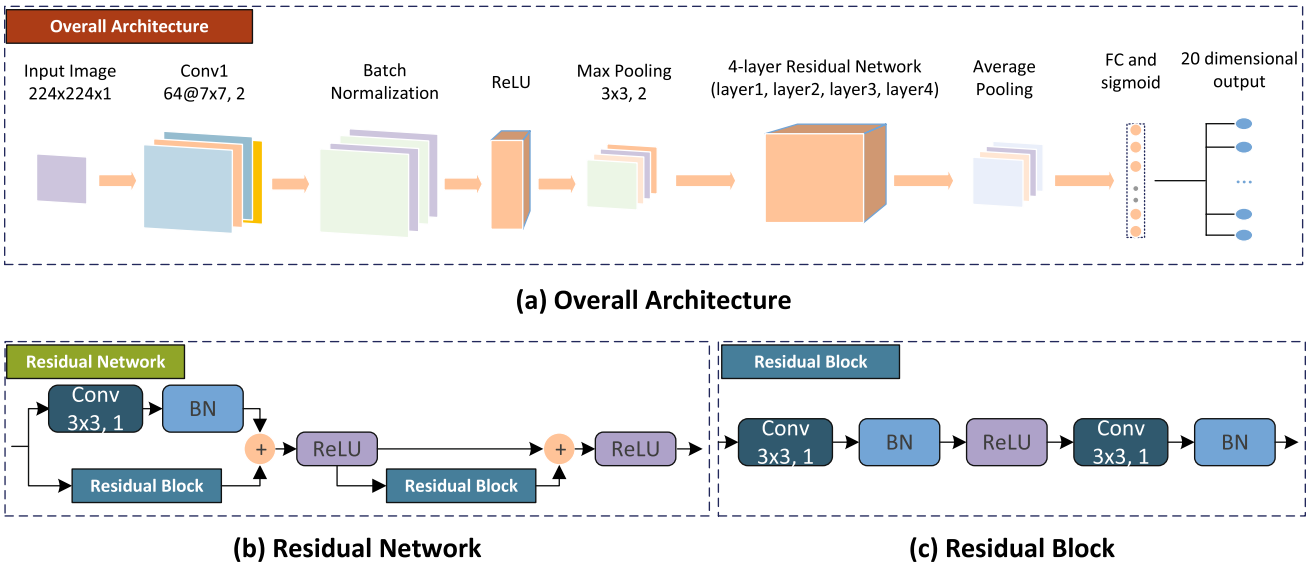


Fig. 5 Modified ResNet-18 architecture diagram

The ZSWMC method uses 20 attributes, as presented in Table 1. These attributes are derived from the 17 attributes presented in Kim and Shim (2024) and are further refined through multiple discussions with wafer fab technicians. We found that the original 17 attributes were insufficient for accurately describing and distinguishing mixed-type defects. To address this limitation, we introduced three additional attributes to capture features that are essential to identify complex mixed-type defects. Thus, our attribute selection is primarily based on domain knowledge, without incorporating additional process information or employing feature selection algorithms.

The 20-dimensional semantic center vectors corresponding to each defect type are detailed in Table 2. For example, the central defect type is characterized by a circular defect pattern (attribute 1) and symmetry (attribute 6), with defective dies clustered in a localized area of the wafer (attribute 4). In addition, the central defect type is typically caused by issues related to chemical mechanical polishing (attribute 13) or deposition (attribute 16) during manufacturing. Consequently, the semantic center vector for the central defect type assigns a value of 1 to attributes 1, 4, 6, 13, and 16, while all remaining attributes are assigned a value of 0.

Model selection and parameter setting

We employ the convolutional layers of *ResNet-18* as the visual feature extractor f, which outputs a 512-dimensional visual feature vector. The overall model architecture is illustrated in Fig. 5a. To accommodate the single-channel input of wafer map data, we modified the first convolutional layer (Conv1) to accept an input format of $224 \times 224 \times 1$. A single fully connected layer serves as the semantic embedder g, mapping the visual feature vectors into a 20-dimensional semantic space to generate corresponding semantic vectors. After the fully connected layer, the function *sigmoid* is used to constrain each element of the semantic vector to the range [0,1]. In Fig. 5a, the residual network contains four layers. The structure of each layer is illustrated in Fig. 5b, and the c presents the structure of residual block.

Parameter settings for the zero-shot learning

Our model was trained on a computational node featuring 8 CPUs, 30 GB of RAM, and an NVIDIA A10 GPU. The Stochastic Gradient Descent (SGD) optimizer was utilized with a batch size of 64, configured with an initial learning rate of 0.01, a momentum of 0.9, and a weight decay(i.e. λ) of 1×10^{-2} to regularize the model and prevent overfitting. Instead of a fixed learning rate, we utilized a cosine annealing scheduler(Loshchilov and Hutter, 2017) (with the cycle half-length $T_{\max}$ set to 100 and the minimum learning rate $\eta_{\min}$ set to 1×10^{-5})over the 100 training epochs for each iteration on the single-type defects dataset D_s. This scheduler dynamically decreases the learning rate from its initial value to nearly zero following a cosine curve, which facilitates more stable convergence and helps escape suboptimal local minima. For updating the semantic center vector, the count of nearest neighbors, represented by m, is 50. These hyperparameters were determined through a literature review and preliminary experiments.

In the progressive pseudo-labeling and retraining strategy, not all mixed-type defect wafer maps are assigned pseudo-labels for model retraining. We select the samples with high-confidence pseudo-labels. For this purpose, we set specific thresholds τ for different mixed-type defects: 0.994

for two-mixed defects, 0.986 for three-mixed defects, and 0.975 for four-mixed defects. We take two-mixed defects as an example to explain how the threshold is determined. After training the model on single-type defects, we classify all mixed-type defects using this model. For each wafer map predicted as two-mixed defects, a cosine similarity score exists between the wafer map and the semantic centroid vector corresponding to the predicted label. We select the top 20% of two-mixed defect wafer maps with the highest similarity scores as pseudo-label samples. The minimum cosine similarity within this subset is set as the threshold for two-mixed defects. After incorporating the two-mixed defect samples with pseudo-labels into model training, we apply the same procedure sequentially to determine threshold values for three-mixed and four-mixed defects.

The confidence threshold decreases from two-mixed to four-mixed defects because it is correlated with the complexity of mixed-type defects. As more single-type defects appear within a mixed defect, the increasing complexity reduces the cosine similarity between a wafer map and its semantic center vector. Since our method selects the top 20% of samples exhibiting the highest similarity within each mixed-type defect category, the minimum similarity (threshold τ) within the selected subset naturally decreases from two-mixed to four-mixed defects.

Parameter settings during the fine-tuning stage

To enable a fair comparison with few-shot learning and self-supervised/semi-supervised approaches, the model is fine-tuned using a small set of mixed-type defect samples with ground-truth labels. Specifically, five samples are randomly chosen from each mixed-type defect category to construct the fine-tuning training set. To mitigate sampling randomness, this fine-tuning procedure is independently repeated five times, with the average performance and standard deviation being reported.

For the fine-tuning configuration, we freeze the weights of the shallow and middle layers of the visual feature extractor f obtained from pre-training on single-type defect data (including Conv1, Layer1, Layer2, and Layer3). Only the Layer4 and the semantic embedder g are fine-tuned. All fine-tuning experiments employ the SGD optimizer with a momentum of 0.9, a weight decay (i.e. λ) of 1×10^{-3}, and a batch size of 16. The initial learning rate is 1×10^{-3}, and a cosine annealing scheduler (the cycle half-length $T_{\max}$ set to 50 and the minimum learning rate $\eta_{\min}$ set to 1×10^{-6}) is used over 50 training epochs. This strategy drives the learning rate to decrease smoothly as training progresses, forcing the model to make only minor parameter updates in the later stages, thus effectively mitigating the risk of overfitting on the small-scale dataset.

Evaluation criteria

In *Generalized zero-shot learning* (GZSL), a unified semantic space encompassing both seen and unseen classes must be constructed, where test samples must be classified within this comprehensive space. This imposes higher demands on the model, requiring simultaneous handling of all seen and unseen classes. GZSL is suited for meeting the demands of open-world applications. In contrast, *Traditional Zero-Shot Learning* (TZSL) constructs a specific semantic space tailored to particular testing tasks. For example, when the test task involves only the detection of single-type defects, the semantic space contains only the semantic center vectors of single-type defects. By narrowing the discrimination scope, TZSL reduces classification difficulty, making it more suitable for validating a model's zero-shot transfer capability in specific constrained scenarios. To evaluate the ZSWMC method, experiments are conducted under both Generalized Zero-Shot Learning (GZSL) and Traditional Zero-Shot Learning (TZSL) settings.

In the experiments, the overall accuracy is adopted as the performance metric for the ZSWMC model:

$$M = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Table 3 Comparison with zero-shot learning methods (mean ± std, %)

Method	GZSL			TZSL		
	M_s	M_{mixed}	H	M_s	M_{mixed}	H
ZSWMC	**97.82±0.15**	**75.09±0.42**	**84.96±0.28**	**98.32±0.12**	**76.67±0.38**	**86.16±0.24**
VCL (Wan et al. 2021)	95.67±0.31	28.34±1.25	43.73±0.87	96.58±0.28	30.28±1.18	46.11±0.79
CDVSc (Wan et al. 2021)	96.21±0.26	33.68±1.12	49.89±0.76	97.39±0.22	36.38±1.05	52.97±0.68
BMVSc (Wan et al. 2021)	96.04±0.29	41.04±0.98	57.51±0.65	97.11±0.25	43.57±0.92	60.15±0.58
WDVSc (Wan et al. 2021)	96.73±0.24	28.82±1.31	44.41±0.91	97.89±0.21	30.73±1.24	46.78±0.83
QFSL (Song et al. 2018)	95.46±0.33	24.55±1.42	39.06±0.95	96.72±0.29	27.17±1.35	42.42±0.86
Bi-VAEGAN (Wang et al. 2023)	97.53±0.17	*60.08±0.85*	*74.36±0.51*	98.02±0.14	*61.74±0.78*	*75.76±0.45*
AD3C-FGN (Zhang et al. 2023)	*97.61±0.16*	42.93±0.92	59.63±0.61	*98.09±0.13*	42.93±0.86	59.72±0.54
TFVAEGAN (Narayan et al. 2020)	94.62±0.35	34.17±1.08	50.21±0.72	96.07±0.31	34.17±1.02	50.41±0.65

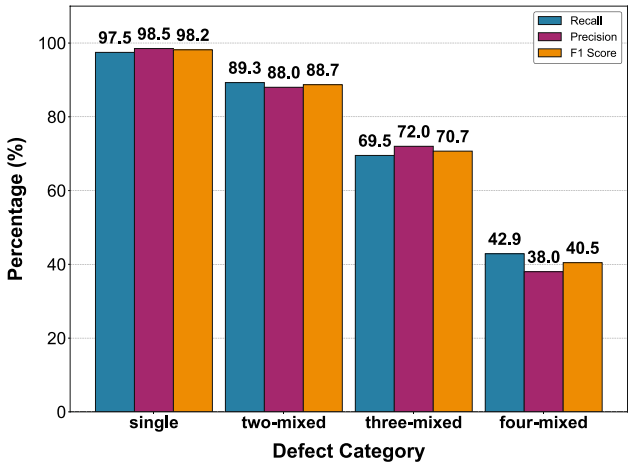


Fig. 6 Impact of defect complexity on performance metrics

- *TP (True Positives)* The count of *positive* instances that the model *accurately* classifies as *positive*.
- *TN (True Negatives)* The count of *negative* instances that the model *accurately* classifies as *negative*.
- *FP (False Positives)* The count of *negative* instances that the model *misclassifies* as *positive*.
- *FN (False Negatives)* The count of *positive* instances that the model *misclassifies* as *negative*.

We compute M_s for the overall accuracy of single-type defects and $\mathrm{M}_t mixed$ for mixed-type defects, using their harmonic mean H as the final performance metric:

$$H = \frac{2 \times \mathrm{M}_s \times \mathrm{M}_{mixed}}{\mathrm{M}_s + \mathrm{M}_{mixed}} \tag{12}$$

Furthermore, to provide a more comprehensive evaluation, the precision, recall, and the F1-score are also used:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

$$F1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

Analysis of experimental results

Comparison and analysis of results

We implemented several representative transductive zero-shot learning methods (Wan et al., 2021; Song et al., 2018; Wang et al., 2023; Zhang et al., 2023; Narayan et al., 2020) and applied them to wafer map classification, comparing their performance with that of the ZSWMC method. In the experiments, all methods treated single-type defect wafer maps as seen classes and mixed-type defects as unseen classes. Each experiment was conducted five times with random initialization of model weights and sample order. Table 3 demonstrates the results of mean ± standard deviation. The best performance is shown in bold, while the second-best is indicated in italics.

The results in Table 3 demonstrate that the ZSWMC method outperforms other methods across every metric. Its superiority is evident from two key statistical perspectives: first, the mean performance of ZSWMC across all metrics is higher than other methods; second, the low standard deviations (e.g., ±0.42% for M_{mixed} under GZSL) observed across all five independent runs indicate that the performance of ZSWMC is stable.

The precision, recall, and F1-score for single-type, two-mixed, three-mixed, and four-mixed defects are presented in Fig. 6. All reported values correspond to the averages obtained from five independent experimental trials. As illustrated, the model achieves performance close to 90% for two-mixed defects, with all three metrics maintained at a high level. This validates the feature generalization capability of the ZSWMC method when handling unseen classes. For three-mixed defects, the model attains an F1-score above 70%. Even in the most challenging scenario involving four-mixed defects, it achieves an F1-score exceeding 40%.

To validate the efficacy of the ZSWMC method, we compare it with several similar approaches (Chiu and Chen, 2021; Kim et al., 2022; Geng et al., 2021; Liang et al., 2024). All comparative results are the mean ± standard deviation over five independent runs. Chiu and Chen (2021) and Kim et al. (2022) also utilize single-type defects as training samples to classify mixed-type defects, which aligns with our zero-shot learning objective. As shown in the first three

Table 4 Comparison with other methods (mean ± std, %)

Method	M_s	M_{mixed}	H
ZSWMC	**97.82±0.15**	**75.09±0.42**	84.96±0.28
Chiu and Chen (2021)	97.73±0.18	68.79±0.65	80.75±0.38
Kim et al. (2022)	96.21±0.25	50.28±0.92	66.04±0.51
ZSWMC (fine-tune)	**98.28±0.12**	**85.72±0.35**	**91.57±0.22**
Geng et al. (2021)	97.43±0.16	82.45±0.48	89.32±0.30
Liang et al. (2024)	94.94±0.28	83.78±0.42	89.01±0.33

Table 5 Performance comparison with semi-supervised/self-supervised learning methods (mean ± std, %)

Method	M_s	M_{mixed}	H
ZSWMC(fine-tune)	**92.8±0.12**	87.72 ±0.53	**97.57±0.22**
Baek et al. (2025)	97.95±0.13	77.52±0.70	86.55±0.47
Kahng and Kim (2021)	98.10±0.11	72.43±0.80	83.33±0.55
Kim and Kang (2021)	98.05±0.12	76.84±0.65	86.18±0.42
Kwak et al. (2023)	97.88±0.13	80.91±0.48	87.98±0.38
Wang et al. (2024)	97.92±0.16	74.15±0.72	84.29±0.51

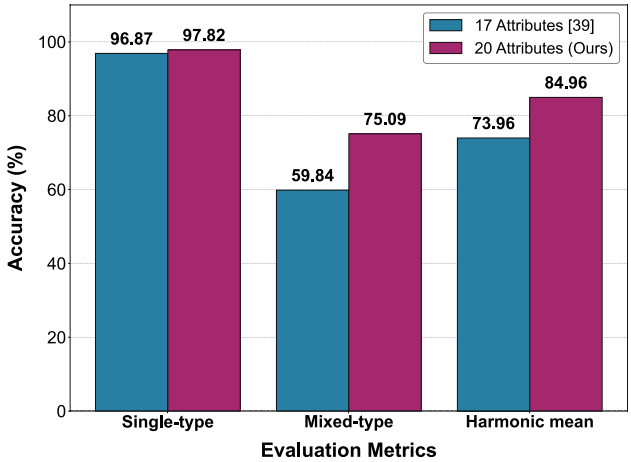


Fig. 7 Impact of attribute on performance metrics

rows of Table 4, when no labeled mixed-type defects are available, the ZSWMC method achieves better performance compared to the methods in Chiu and Chen (2021) and Kim et al. (2022).

Because the ZSWMC method do not require any labeled mixed-type defect samples, its classification accuracy is lower than that of few-shot learning approaches in Geng et al. (2021) and Liang et al. (2024). However, when supplemented with a small number of labeled mixed-type defect samples, the ZSWMC method outperforms few-shot learning methods in classification accuracy.

The comparison results between ZSWMC and the methods in Geng et al. (2021) and Liang et al. (2024) are shown in the last three rows of Table 4. For fair comparison, our ZSWMC method and the compared methods were evaluated using five labeled samples per mixed-type defect as training samples. To indicate whether mixed-type defect samples are included during training, we denote the method without such samples as ZSWMC, and the method that incorporates mixed-type defect samples as ZSWMC (fine-tune).

The results demonstrate that the M_{mixed} of ZSWMC is $85.72 \pm 0.35\%$, surpassing the few-shot learning methods Geng et al. (2021); Liang et al. (2024). Furthermore, compared to the accuracy of 75.09% achieved by the ZSWMC without labeled mixed-type defect samples, the ZSWMC utilizing few-shot learning improves accuracy by 10.63%. These results indicate that the ZSWMC method can significantly improve classification performance with minimally labeled samples.

For a comprehensive assessment of the ZSWMC(fine-tune) method, a comparative analysis is conducted against several state-of-the-art self-supervised and semi-supervised learning approaches (Baek et al., 2025; Kahng and Kim, 2021; Kim and Kang, 2021; Kwak et al., 2023; Wang et al., 2024). For fairness, all methods are trained on the same set of samples.

The results are summarized in Table 5. All reported values are expressed as the mean ± standard deviation, where the best results are emphasized in bold. For single-type defects recognition, all methods exhibited comparable performance because they were trained on single-type defect samples. For the recognition of mixed-type defects, the ZSWMC(fine-tune) method achieved an accuracy of 85.72%, significantly outperforming the other methods by margins ranging from 4.81% (vs. Kwak et al. (2023)) to 13.29% (vs. Kahng and Kim (2021)).

In terms of the overall performance metric H, the ZSWMC(fine-tune) method ranked first with a score of 91.57%, benefiting from its superior performance on both single-type and mixed-type defects. Among the remaining techniques, Kwak et al. (2023) obtained the highest overall score of 87.98% due to its relatively better performance on M_{mixed}, although it still exhibited a gap of 3.59% compared

Table 6 Ablation study on EECO, SU, and PL

EECO	SU	PL	M_{all}	M_{mixed}	M_s	M_2	M_3	M_4
✗	✗	✗	48.62	27.91	97.69	44.35	15.83	8.50
✓	✗	✗	49.69 (+1.07)	31.57 (+3.66)	97.13 (−0.56)	52.88 (+8.53)	16.58 (+0.75)	8.82 (+0.32)
✗	✓	✗	53.52 (+4.90)	40.27 (+12.36)	96.28 (−1.41)	60.72 (+16.37)	27.26 (+11.43)	14.52 (+6.02)
✗	✗	✓	50.12 (+1.50)	29.76 (+1.85)	97.42 (−0.27)	47.82 (+3.47)	17.57 (+1.74)	10.41 (+1.91)
✓	✓	✗	62.96 (+14.34)	53.47 (+25.56)	96.95 (−0.74)	68.23 (+23.88)	44.33 (+28.50)	28.50 (+20.00)
✓	✗	✓	51.33 (+2.71)	35.17 (+7.26)	97.09 (−0.6)	58.26 (+13.91)	25.43 (+9.6)	15.64 (+7.14)
✓	✓	✓	**80.31** (+31.69)	**75.09** (+47.18)	**97.82** (+0.13)	**89.42** (+45.07)	**69.42** (+53.59)	**43.25** (+34.75)

to the ZSWMC(fine-tune) method. These results indicate that, even with only a small number of mixed-type defect samples, the ZSWMC (fine-tune) method preserves strong single-type defects recognition performance while achieving a clear advantage in mixed-type defects recognition.

To evaluate the presented 20 attributes, we compared them with the original 17 attributes from Kim and Shim (2024) under identical experimental conditions. As shown in Fig. 7, our method shows significant improvements in all evaluation metrics. The accuracy for single-type defects(M_s) increased from 96.87% to 97.82%. The most notable improvement was observed in mixed-type defects, where the accuracy (M_{mixed}) rises from 59.84% to 75.09%, yielding a gain of over 15 percentage points. The harmonic mean (H) also improves from 73.96 to 84.96%. These results demonstrate that the extended attributes capture more discriminative features for mixed-type defects.

Ablation study

To ensure the statistical reliability of the results, all ablation studies reported in this section were conducted over five independent runs. The reported results are the mean of these five runs.

Component analysis

We conduct a comprehensive ablation study under GZSL to evaluate the contributions of the three proposed optimization strategies: *End-to-End Collaborative Optimization* (EECO), *Semantic Space Update* (SU), and *Progressive Pseudo-Labeling* (PL). A baseline model incorporating none of these strategies serves as our starting point, with its results detailed in the first row of Table 6. The six M metrics from left to right correspond to test data encompassing all defects, mixed-type defects, single-type, two-mixed, three-mixed, and four-mixed defects.

The experimental results reveal several key insights. First, while the integration of all three strategies yields a substantial performance gain for mixed-type defects, their individual application leads to a slight decrease in(M_s). This is due to a shift in the feature-learning objective. The baseline model optimizes for discriminative features directly aligned with single-type ground-truth labels. In contrast, the introduction of EECO, SU, or PL guides the model to learn more localized and interpretable attributes that enhance cross-category generalization, which may come at a minor cost to the sharp inter-class separation beneficial for single-type recognition.

Second, all three strategies contribute significantly to the classification of mixed-type defects (M_{mixed}), with which the baseline model struggles. When applied individually, EECO, SU, and PL improve M_{mixed} by 3.66%, 12.36%, and 1.85%, respectively. Their combination is most effective, resulting in a notable improvement of 47.18% over baseline.

Third, the SU strategy emerges as the single most impactful component. Its single application provides the largest individual gain (12.36% in M_{mixed}). Furthermore, adding SU to the combination of EECO and PL brings an additional 39.92% improvement in M_{mixed}.

Finally, the PL strategy alone shows a limited effect (1.85% improvement). This is likely because, without the robust feature representations enabled by EECO and SU, the initial model generates noisy pseudo-labels, which in turn provide limited guidance for effective model refinement.

Moreover, to validate the effectiveness of a single FC layer as the semantic embedder g, we compared the classification performance of semantic embedders with different depths. Specifically, we evaluated three architectures: (1) a single FC layer (512→20), (2) two FC layers (512→

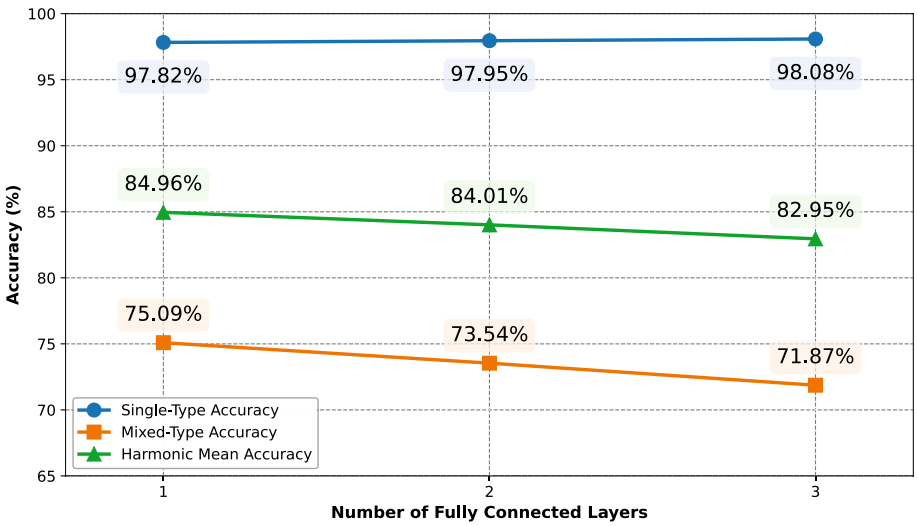


Fig. 8 Effect of semantic embedder depth

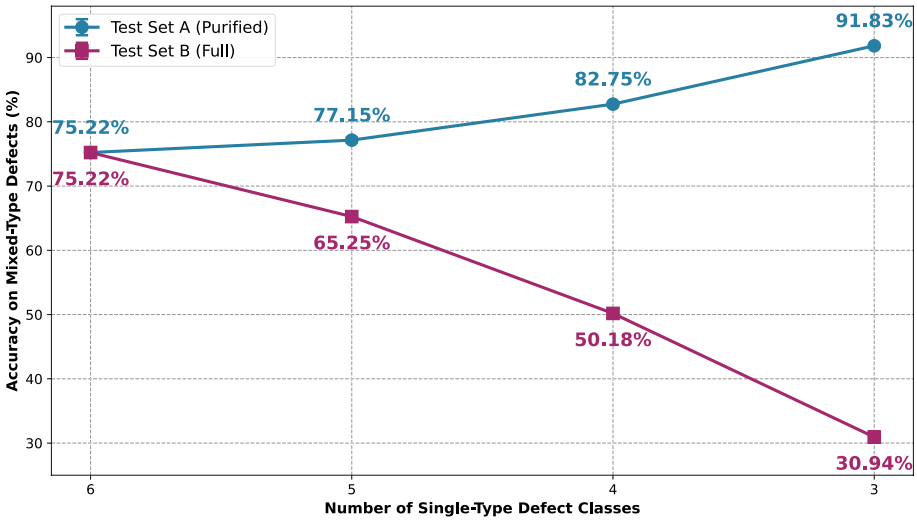


Fig. 9 Effect of single-type defect categories

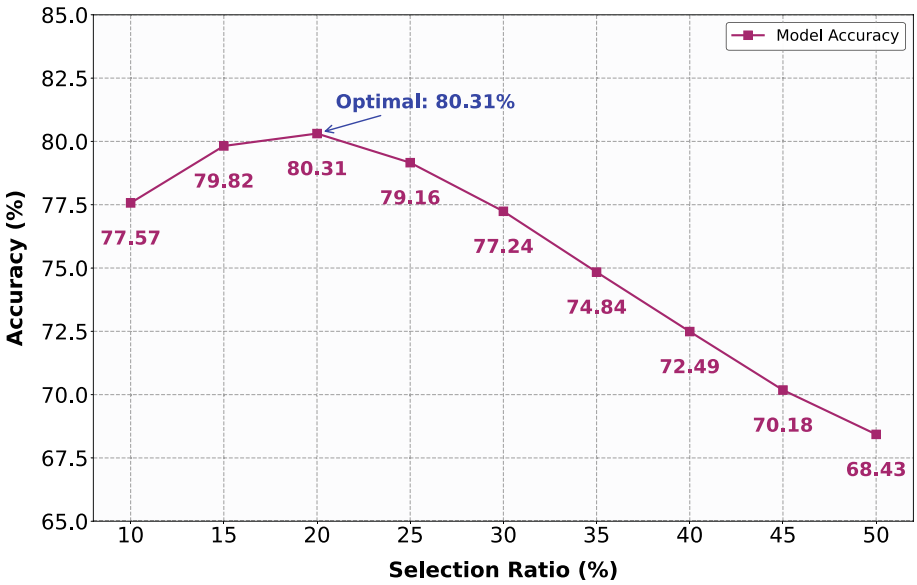


Fig. 10 Impact of pseudo-label selection ratio

256→20), and (3) three FC layers (512→256→128→20). All models were trained in identical settings and evaluated in the same test set. The corresponding results are presented in Fig. 8.

The experimental results demonstrate that although increasing the depth of the semantic embedder slightly improves performance on single-type defects—likely due to overfitting to their data distribution—it substantially degrades generalization to mixed-type defects. This is evidenced by a clear decline in mixed-type defect accuracy and harmonic mean accuracy as network depth increases. Thus, employing deeper FC layers as the semantic embedder can affect the model's capability in classifying mixed-type defects. In contrast, a single FC layer achieves the best overall generalization, maintaining high single-type defect recognition accuracy while delivering superior performance on mixed-type defects, thereby confirming its effectiveness as the semantic embedder for our task.

To evaluate how the sample size of single-type defects affects the recognition performance for mixed-type defects, we conducted an ablation study. Although the original dataset contains nine single-type defects, we found that the `Normal`, `Nearfull`, and `Random` categories do not serve as constituent components of any mixed-type defects. Therefore, we removed these three single-type defects from the training set, retained the remaining six, and further examined how reducing the number of single-type defect categories influences performance.

The experiment used two different test sets. Test Set A (Purified Set), which contains only mixed-type defects whose constituent single-type defects all appear in the training set. Test Set B (Full Set) includes all 29 mixed-type defect categories. For example, if the training samples

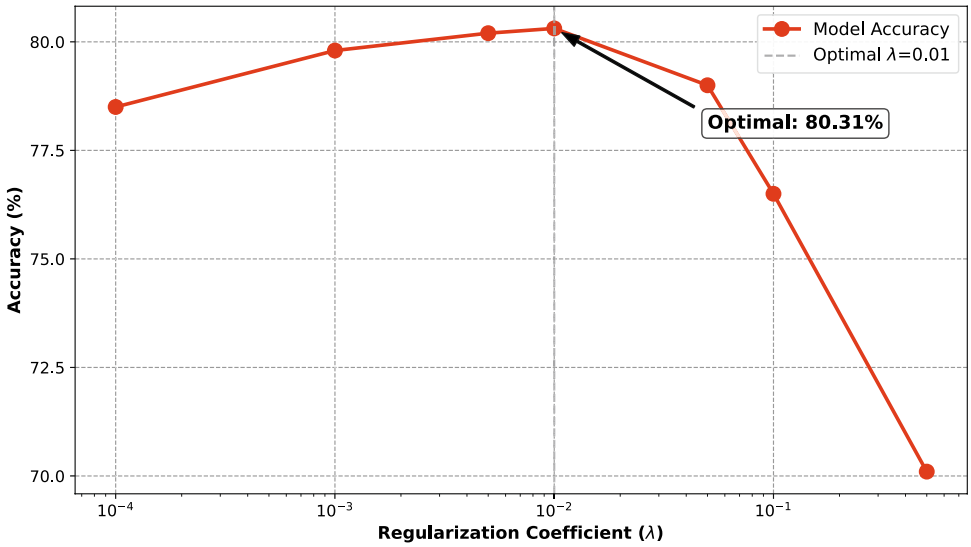


Fig. 11 Impact of regularization coefficient (λ)

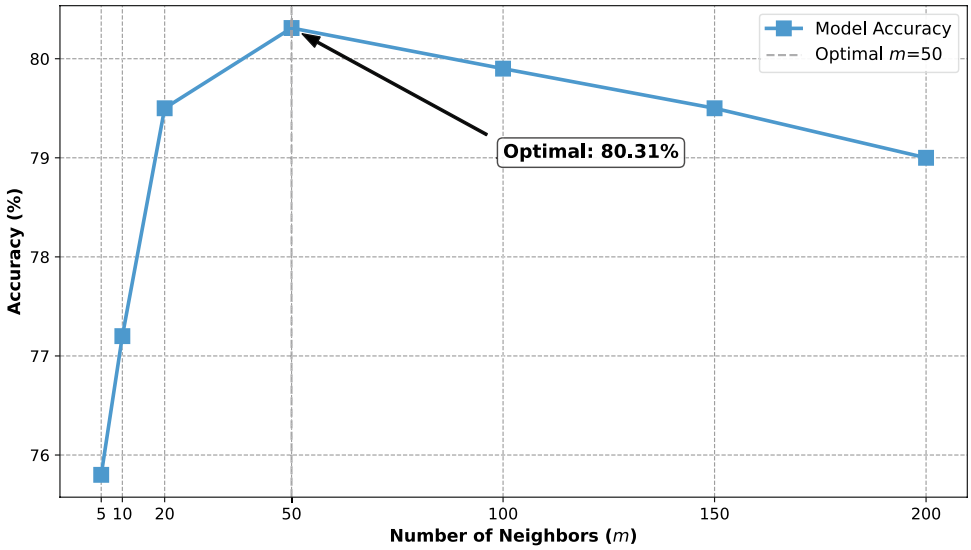


Fig. 12 Impact of neighbor count (m)

include only the `Center`, `Loc`, and `Scratch` single-type defects, then Test Set A will consist of the four corresponding mixed-type defect categories: `C+L`, `C+S`, `L+S` and `C+L+S`.

Figure 9 demonstrates the influence of single-type defect categories on model performance. The six single-type defect categories referred to in the experiments are `Center`, `Loc`, `Donut`, `Scratch`, `Edge_loc`, and `Edge_ring`. When constructing training sets with 5, 4, and 3 single-type defect categories, we employed a random discarding strategy and averaged the results over five independent random experiments to ensure statistical reliability.

For the scenario with 5 single-type defect categories, we respectively discarded one category in each of the five experiments: `Center`, `Donut`, `Loc`, `Edge_ring`, `Scratch`.

For the scenario with 4 single-type defect categories, we randomly selected two categories to discard. The five discard combinations were: (1) `Center`, `Donut`; (2) `Edge_loc`, `Loc`; (3) `Edge_ring`, `Scratch`; (4) `Center`, `Edge_loc`; (5) `Donut`, `Scratch`.

For the scenario with 3 single-type defect categories, we randomly selected three categories to discard. The five combinations of discards were: (1) `Center`, `Edge_loc`, `Donut`; (2) `Loc`, `Scratch`, `Edge_ring`; (3) `Center`,`Scratch`, `Edge_ring`; (4) `Donut`, `Loc`, `Edge_loc`; (5) `Center`, `Loc`, `Scratch`.

On Test Set A (Purified Set), as the number of training categories decreased from six to three, the model accuracy significantly increased from 75.22 to 91.83%. This improvement indicates that within a constrained task scope, reducing the defect categories enables the model to concentrate more effectively on learning discriminative features, thereby improving classification performance.

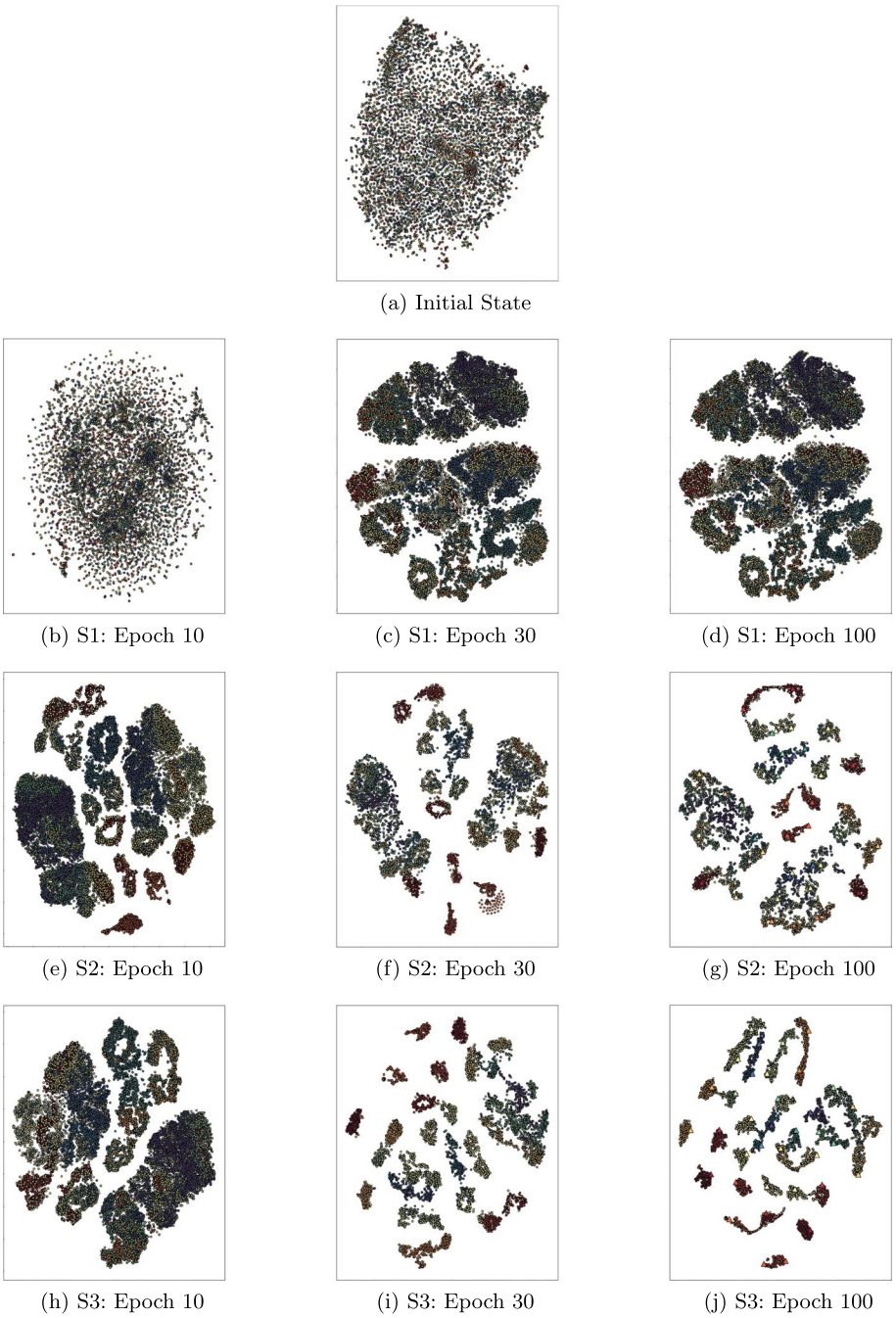
A different trend was observed on Test Set B (Full Set), which highlights the limitation of the ZSWMC method.

When the number of training defect categories decreased from six to three, the model accuracy dropped from 75.22 to 30.94%. This result illustrates a key premise of our approach: the model's ability to recognize mixed-type defects is fundamentally bounded by its knowledge of single-type defects. It can accurately identify mixed-type defects composed solely of known single-type defects, but it cannot generalize to those containing unknown.

Sensitivity analysis of hyperparameters

In the PL strategy, the top 20% of wafer maps with the highest similarity scores are assigned pseudo-labels for model retraining. To verify the optimality of the 20% selection ratio, experiments were performed using selection ratios varying from 10 to 50%. The results are shown in Fig 10.

As shown in the figure, the model accuracy (M_{all}) increased steadily as the selection ratio rose from 10% to 15%, reaching a peak accuracy of 80.31% at 20%. Beyond this point, increasing the selection ratio led to a notable



(a) Initial State

(b) S1: Epoch 10

(c) S1: Epoch 30

(d) S1: Epoch 100

(e) S2: Epoch 10

(f) S2: Epoch 30

(g) S2: Epoch 100

(h) S3: Epoch 10

(i) S3: Epoch 30

(j) S3: Epoch 100

Fig. 13 Visualization of multi-stage semantic space evolution

decline in performance, with accuracy dropping to 79.16% at 25%, 77.24% at 30%, and eventually to 68.43% at 50%. This degradation occurs because the incorporation of too many pseudo-labeled samples lowers the reliability of the label and introduces additional noise into the training process. These results confirm that a 20% selection ratio provides the optimal accuracy in our method.

We evaluated the model's sensitivity to two key hyperparameters: the regularization coefficient λ and the number of neighbors m. Figure 11 illustrates how the accuracy M_{all} varies with λ. The model exhibits a clear inverted-U relationship with respect to λ. When λ is too small (e.g., 0.0001), weak regularization cannot prevent overfitting to noise in the training data, restricting the model's capacity to generalize to mixed-type defects. As λ increases to the optimal value of 0.01, the model reaches its peak performance of 80.31%. However, when λ becomes excessively large (e.g., 0.1), the overly strong regularization hinders effective learning, causing underfitting and a corresponding performance drop. These findings underscore the necessity of appropriately adjusting the regularization hyperparameter in the proposed method.

Figure 12 illustrates the influence of the neighbor counts m on M_{all} during the update of the semantic center vectors. When m is small (e.g., 5 or 10), the limited number of samples used to compute the semantic center vectors makes the results susceptible to noise and instability, leading to low accuracy. As m increases to the optimal value of 50, the semantic center vectors become more reliable, better capturing the essential characteristics of each defect type and achieving the maximum accuracy of 80.31%. It should be noted that M_{all} remains above 79% in a broad range of $m = 20$ to $m = 200$, demonstrating the substantial robustness of our method to the choice of m.

Comprehensive sensitivity analyses on the regularization coefficient λ and the number of neighbors m reveal that our method exhibits favorable properties around its key hyperparameters. Both the pseudo-label ratio and λ exhibit a distinct "inverted-U" sensitivity, confirming that high-quality pseudo-label filtering and an appropriate regularization strength are crucial for optimal performance. Meanwhile, the model maintains stable performance across a broad range of m values, demonstrating the robustness of the semantic center updating strategy.

Table 7 Quantitative evaluation of clustering quality

Strategy	Metric 1 ↓	Metric 2 ↑	Metric 3 ↑
S1 (EECO only)	0.352	0.418	0.191
S2 (EECO + SU)	0.241	0.583	0.428
S3 (EECO + SU + PL)	0.158	0.724	0.613

Visualization and semantic correlation analysis

Visualization analysis of semantic space

The visualization analysis was performed to intuitively demonstrate the performance of the model from different perspectives. Figure 13 shows the t-SNE visualizations comparing three strategies:S1 (EECO only), S2 (EECO + SU), and S3 (EECO + SU + PL). We visualize checkpoints at training epochs 10, 30 and 100 for each strategy.

In Fig. 13, the differently colored triangles (▲) represent distinct semantic center vectors, while the circles (●) in different colors indicate different semantic vectors. The Figure clearly demonstrates the dynamic optimization of the semantic space under different strategies. The semantic vectors in the initial state (Fig. 13a) exhibit an unstructured, random distribution, indicating that the model has not yet learned discriminative features.

Under S1 (EECO only), the samples gradually aggregate by epoch 10 (Fig. 13b). As training progresses (Fig. 13c and d), several large clusters emerge, but the category boundaries remain blurred, and the intra-class structure is loose. This reveals the limitation of S1 in achieving fine-grained classification.

S2(EECO + SU) brings about a remarkable improvement. By epoch 10 (Fig. 13e), it already exhibits tighter clustering than S1. By epoch 30 (Fig. 13f), the inter-class separation and intra-class cohesion increase noticeably. By epoch 100 (Fig. 13g), the semantic space is well structured with clear decision boundaries, demonstrating the benefit of updating semantic center vectors.

S3 (EECO + SU+PL) produces even faster clustering and clearer inter-class separation by epoch 10 (Fig. 13h) and epoch 30 (Fig. 13i). The final state (Fig. 13j) displays highly cohesive clusters. Compared to S2 (Fig. 13g), the class clusters under S3 are internally more compact, and the margins between clusters are larger and more distinct. This indicates that the supervisory signals from pseudo-labels guide the model to learn more refined semantic features, thereby improving the overall semantic-space structure and yielding superior classification performance.

To quantitatively evaluate the semantic space under different strategies, we compute three clustering metrics:

- *Metric 1: Intra-class cohesion* Measures intra-class compactness, computed as the average cosine distance between all sample pairs within a class. A lower value indicates higher similarity and better cohesion within the class.

Table 8 Semantic correlation analysis

Defect categories	C	D	EL	L	ER	S
Two-mixed						
C + EL	0.711	0.200	0.764	0.537	0.501	0.446
C + ER	0.677	0.461	0.513	0.289	0.795	0.304
C + L	0.750	0.203	0.436	0.814	0.174	0.475
C + S	0.741	0.204	0.441	0.566	0.297	0.739
D + EL	0.315	0.632	0.771	0.412	0.722	0.321
D + ER	0.204	0.695	0.473	0.069	0.931	0.215
D + L	0.491	0.392	0.469	0.856	0.199	0.343
D + S	0.369	0.402	0.359	0.352	0.338	0.845
EL + L	0.335	0.053	0.810	0.811	0.420	0.463
EL + S	0.335	0.053	0.813	0.568	0.422	0.738
ER + L	0.316	0.484	0.413	0.413	0.836	0.313
ER + S	0.332	0.501	0.439	0.189	0.873	0.470
L + S	0.479	0.065	0.555	0.816	0.189	0.741
Three-mixed						
C + EL + L	0.638	0.224	0.710	0.689	0.496	0.538
C + EL + S	0.645	0.198	0.706	0.590	0.486	0.642
C + ER + L	0.617	0.437	0.495	0.500	0.729	0.422
C + ER + S	0.624	0.465	0.500	0.385	0.776	0.410
C + L + S	0.680	0.218	0.523	0.728	0.314	0.659
D + EL + L	0.395	0.547	0.687	0.663	0.656	0.400
D + EL + S	0.412	0.544	0.684	0.502	0.638	0.614
D + ER + L	0.340	0.595	0.415	0.435	0.774	0.341
D + ER + S	0.355	0.645	0.417	0.210	0.825	0.440
D + L + S	0.550	0.366	0.528	0.721	0.321	0.666
EL + L + S	0.434	0.101	0.738	0.741	0.413	0.665
ER + L + S	0.445	0.467	0.416	0.430	0.785	0.440
Four-mixed						
C + EL + L + S	0.607	0.238	0.662	0.664	0.500	0.623
C + ER + L + S	0.624	0.421	0.464	0.500	0.708	0.509
D + EL + L + S	0.452	0.505	0.618	0.628	0.591	0.584
D + ER + L + S	0.406	0.575	0.429	0.492	0.753	0.497

$$\text{Cohesion}(C_k) = \frac{1}{|C_k| \cdot (|C_k| - 1)} \sum_{\substack{i,j \in C_k \\ i \neq j}} \text{CosineDistance}(\text{a}_i, \text{a}_j) \tag{16}$$

where C_k denotes the *k*-th class ($k = 1, 2, \ldots, K$, K is the total number of classes), $|C_k|$ is the number of samples in C_k ($|C_k| \geq 2$), a_i and a_j are the semantic vectors of the *i*-th and *j*-th samples in C_k ($i \neq j$). The cosine distance is defined as $\text{CosineDistance}(\cdot, \cdot) = 1 - \text{CosineSimilarity}(\cdot, \cdot)$, which ranges from 0 to 2.

- *Metric 2: Inter-class separation* Measures the separation between distinct classes, defined as the mean cosine distance between the centroids of all class pairs. A higher value indicates clearer boundaries between classes.

$$\text{Separation} = \frac{2}{K \cdot (K-1)} \sum_{k=1}^{K} \sum_{l=k+1}^{K} \text{CosineDistance}(\boldsymbol{\mu}_k, \boldsymbol{\mu}_l) \tag{17}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_l$ are the mean semantic vectors for classes C_k and C_l,respectively. The centroid for a given class C_k is calculated as:$\boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{i \in C_k} \text{a}_i$, where a_i is the semantic vector of the *i*-th sample.

- *Metric 3: silhouette coefficient* A comprehensive metric combining intra-class cohesion with inter-class separation, with a range of $[-1, 1]$. A value nearer to 1 signifies superior clustering quality.

$$\text{Silhouette} = \frac{1}{N} \sum_{i=1}^{N} s(i), \quad s(i) = \frac{d_{\text{inter}}(i) - d_{\text{intra}}(i)}{\max\{d_{\text{intra}}(i), d_{\text{inter}}(i)\}} \tag{18}$$

where N denotes the total sample count in dataset, $s(i)$ represents the silhouette coefficient for the *i*-th sample. The $d_{\text{intra}}(i)$ is the average cosine distance between sample *i* and all other samples within its own class C_k, while $d_{\text{inter}}(i)$ is the minimum average cosine distance from sample *i* to samples in any other class. The $\max\{\cdot, \cdot\}$ term in the denominator ensures $s(i)$ falls within the interval $[-1, 1]$. A value

approaching 1 indicates that the sample is well aligned with its assigned class and clearly separated from others.

The results for the three strategies (S1–S3 at Epoch 100) are presented in Table 7. Metric 1 reports the average Intra-class Cohesion across all defect categories. From S1 to S3, the Metric 1(Intra-class Cohesion) consistently decreases ($0.352 \rightarrow 0.241 \rightarrow 0.158$), indicating that each successive strategy effectively drives samples of the same class to cluster more tightly. Meanwhile, the Metric 2(Inter-class Separation) increases markedly ($0.418 \rightarrow 0.583 \rightarrow 0.724$), showing that class boundaries become progressively more distinct. The combined improvements of these two metrics are reflected in the steady rise of Metric 3 (Silhouette Coefficient) from 0.191 to 0.613.

Specifically, S2 (EECO+SU) yields substantial improvements across all metrics compared to S1 (EECO only), confirming that updating semantic center vectors effectively guides feature learning and enhances the structure of the semantic space. S3 (EECO+SU+PL) achieves the best overall performance. Its minimal Intra-class Cohesion coupled with maximal Inter-class Separation and Silhouette Coefficient indicate that the additional supervisory signals from pseudo-labels allow the model to acquire more discriminative features from mixed-type defects. The high correlation observed between the quantitative metrics and the classification accuracy validates the efficacy of the ZSWMC method.

Semantic correlation analysis

To verify that the ZSWMC method can effectively learn the semantic information of mixed-type defects from single-type defects, we conducted a semantic correlation analysis. The semantic correlation score between a mixed-type defect category y_i and a single-type defect category y_j is defined as follows: for each wafer map labeled as y_i, its semantic vector is first derived using the ZSWMC method. Subsequently, the cosine similarity between the semantic vector and the semantic center vector of the single-type defect category y_j is computed. The average cosine similarity across all samples in y_i is used as the semantic correlation score. A higher score indicates a stronger semantic correlation.

Table 8 presents the results of semantic correlation analysis. All mixed-type defects are listed in the first column, and the remaining columns correspond to the six single-type defects. The results show that each mixed-type defect exhibits the highest semantic correlation with its constituent single-type defects. For example, the two-mixed defect `C+EL` achieves scores of 0.711 with `C` and 0.764 with `EL`, which are substantially higher than its correlation with non-constituent defects such as `D` (0.200). This pattern consistently holds for three-mixed (e.g., `C+EL+L`) and four-mixed defects (e.g., `C+EL+L+S`), confirming that the ZSWMC method effectively captures the semantic information of mixed-type defects from single-type defects.

These results also reveal the relationship between the complexity of mixed-type defects and their constituent single-type defects: the semantic correlation score decreases as the number of constituent defects increases. Two-mixed defects generally exhibit high correlation scores with their corresponding single-type defects, typically above 0.7; three-mixed defects fall mainly within the 0.6–0.8 range; and four-mixed defects drop further to the 0.5–0.7 range. This trend explains the model's progressively declining classification performance from two-mixed to four-mixed defects.

Conclusion

This paper presents a wafer map classification method based on transductive zero-shot learning, referred to as the ZSWMC method. This method utilizes labeled single-type defect wafer maps to classify mixed-type defects, effectively reducing the annotation costs for mixed-type defects. Before training, semantic center vectors are constructed based on prior knowledge, serving as both training targets for the model and the basis for classification. The training phase employs an end-to-end approach to collaboratively optimize the visual feature extractor and semantic embedder, enhancing their ability to capture intrinsic relationships between visual features and semantic information. Through iteratively updating semantic center vectors, the ZSWMC method improves the alignment between the visual and semantic spaces. Additionally, a progressive pseudo-labeling and retraining strategy is adopted to iteratively incorporate information from mixed-type defects, further improving the model's generalization capability.

The ZSWMC method achieves superior classification accuracy for both seen and unseen classes compared to other transductive zero-shot learning methods. The ablation study highlights the significant advantages of the EECO, SU, and PL optimization strategies.

Although the ZSWMC method exhibits excellent performance for most mixed-type defects, its classification accuracy for the four-mixed defect class remains relatively low. This limitation stems primarily from the limited precision of manually defined semantic attributes, which are insufficient to capture the complex interrelationships present in four-mixed defects. To overcome this limitation, future research could explore unsupervised attribute-learning approaches, such as variational autoencoders or deep clustering, to automatically derive more accurate semantic attributes for mixed-type defects and further enhance model performance.

Acknowledgements This research was supported in part by the National Natural Science Foundation of China under Grant 62174048, and Grant 62027815.

Author contributions Jun Liu: Methodology, Software, Original Draft. Jifei Lu: Experiments, Validation, Analysis, Original Draft. Tian Chen: Resources, Supervision, Project administration, Review and Editing. Xi Wu: Experiments, Review and Editing. Huaguo Liang: Resources, Supervision, Project administration, Review and Editing. Xiaohui Yuan: Supervision, Review and Editing. Yen Pham: Experiments, Original Draft. All authors have reviewed and approved the final version of this manuscript.

Data availability The wafer map dataset used in this study is a publicly available dataset from MixedWM38, and can be accessed at https://www.kaggle.com/datasets/co1d7era/mixedtype-wafer-defect-datasets/data.

Code availability The core code of our method is publicly available at https://github.com/Luufeii/ZSWMC.git..

Declarations

Conflict of interest 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.

References

Alawieh, M. B., Boning, D., & Pan, D. Z. (2020). Wafer map defect patterns classification using deep selective learning. In *Proceedings of ACM/IEEE design automation conference (DAC)*. pp. 1–6. https://doi.org/10.1109/DAC18072.2020.9218580

Baek, I., Hwang, S. J., & Kim, S. B. (2025). CowSSL: Contrastive open-world semi-supervised learning for wafer bin map. *Journal of Intelligent Manufacturing,36*(3), 2163–2175. https://doi.org/10.1007/s10845-024-02351-0

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation,15*(6), 1373–1396. https://doi.org/10.1162/089976603321780317

Cheng, K. C.-C., Li, K. S.-M., Wang, S.-J., Huang, A. Y.-A., Lee, C.-S., Chen, L. L.-Y., Liao, P. Y.-Y., & Tsai, N. C.-Y. (2022). Wafer defect pattern classification with explainable-decision tree technique. In *2022 IEEE International Test Conference (ITC)*, pp. 549–553. https://doi.org/10.1109/ITC50671.2022.00070

Chen, S., Huang, Z., Wang, T., Hou, X., & Ma, J. (2024). Mixed-type wafer defect detection based on multi-branch feature enhanced residual module. *Expert Systems with Applications,242*, 122795. https://doi.org/10.1016/j.eswa.2023.122795

Chen, S., Liu, M., Hou, X., Zhu, Z., Huang, Z., & Wang, T. (2023). Wafer map defect pattern detection method based on improved attention mechanism. *Expert Systems with Applications,230*, 120544. https://doi.org/10.1016/j.eswa.2023.120544

Chen, S., Zhang, Y., Hou, X., Shang, Y., & Yang, P. (2022). Wafer map failure pattern recognition based on deep convolutional neural network. *Expert Systems with Applications,209*, 118254. https://doi.org/10.1016/j.eswa.2022.118254

Chiu, M.-C., & Chen, T.-M. (2021). Applying data augmentation and mask R-CNN-based instance segmentation method for mixed-type wafer maps defect patterns classification. *IEEE Transactions on Semiconductor Manufacturing,34*(4), 455–463. https://doi.org/10.1109/TSM.2021.3118922

Feng, K., Ji, J. C., Ni, Q., & Beer, M. (2023). A review of vibration-based gear wear monitoring and prediction techniques. *Mechanical Systems and Signal Processing,182*, 109605. https://doi.org/10.1016/j.ymssp.2022.109605

Feng, K., Ji, J. C., Zhang, Y., Ni, Q., Liu, Z., & Beer, M. (2023). Digital twin-driven intelligent assessment of gear surface degradation. *Mechanical Systems and Signal Processing,186*, 109896. https://doi.org/10.1016/j.ymssp.2022.109896

Geng, H., Sun, Q., Chen, T., Xu, Q., Ho, T.-Y., & Yu, B. (2023). Mixed-type wafer failure pattern recognition. In *Proceedings of IEEE Asia and South Pacific design automation conference (ASP-DAC)*, pp. 727–732. https://doi.org/10.1145/3566097.3568363

Geng, H., Yang, F., Zeng, X., & Yu, B. (2021). When wafer failure pattern classification meets few-shot learning and self-supervised learning. In *Proceedings of IEEE/ACM international conference on computer aided design (ICCAD)*, pp. 1–8. https://doi.org/10.1109/ICCAD51958.2021.9643518

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

Hyun, Y., & Kim, H. (2020). Memory-augmented convolutional neural networks with triplet loss for imbalanced wafer defect pattern classification. *IEEE Transactions on Semiconductor Manufacturing,33*(4), 622–634. https://doi.org/10.1109/TSM.2020.3010984

Jang, S.-J., Kim, J.-S., Kim, T.-W., Lee, H.-J., & Ko, S. (2019). A wafer map yield prediction based on machine learning for productivity enhancement. *IEEE Transactions on Semiconductor Manufacturing,32*(4), 400–407. https://doi.org/10.1109/TSM.2019.2945482

Jang, J., & Lee, G. T. (2023). Decision fusion approach for detecting unknown wafer bin map patterns based on a deep multitask learning model. *Expert Systems with Applications,215*, 119363. https://doi.org/10.1016/j.eswa.2022.119363

Jang, J., Seo, M., & Kim, C. O. (2020). Support weighted ensemble model for open set recognition of wafer map defects. *IEEE Transactions on Semiconductor Manufacturing,33*(4), 635–643. https://doi.org/10.1109/TSM.2020.3012183

Kahng, H., & Kim, S. B. (2021). Self-supervised representation learning for wafer bin map defect pattern classification. *IEEE Transactions on Semiconductor Manufacturing,34*(1), 74–86. https://doi.org/10.1109/TSM.2020.3038165

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. https://doi.org/10.1007/s10845-022-01994-1

Kim, D., & Kang, P. (2021). Dynamic clustering for wafer map patterns using self-supervised learning on convolutional autoencoders. *IEEE Transactions on Semiconductor Manufacturing,34*(4), 444–454. https://doi.org/10.1109/TSM.2021.3107720

Kim, T. S., Lee, J. W., Lee, W. K., & Sohn, S. Y. (2022). Novel method for detection of mixed-type defect patterns in wafer maps based on a single shot detector algorithm. *Journal of Intelligent Manufacturing,33*(6), 1715–1724. https://doi.org/10.1007/s10845-021-01755-6

Kim, H. K., & Shim, J. (2024). Generalized zero-shot learning for classifying unseen wafer map patterns. *Engineering Applications of Artificial Intelligence,133*, 108476. https://doi.org/10.1016/j.engappai.2024.108476

Kwak, M. G., Lee, Y. J., & Kim, S. B. (2023). SWaCo: Safe wafer bin map classification with self-supervised contrastive learning. *IEEE Transactions on Semiconductor Manufacturing,36*(3), 416–424. https://doi.org/10.1109/TSM.2023.3280891

Lampert, C. H., Nickisch, H., & Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*. pp. 951–958. https://doi.org/10.1109/CVPR.2009.5206594

Li, J., Tang, H., Tang, D., & Yang, Z. (2023). Multi-label zero-shot learning for industrial fault diagnosis. In *2023 6th international conference on information communication and signal processing (ICICSP)*. pp. 1235–1240 . https://doi.org/10.1109/ICICSP59554.2023.10390617

Liang, Q., Zhou, J., & Wang, Y. (2024). Masked autoencoder with dynamic multi-loss adaptation mechanism for few shot wafer map pattern recognition. *Engineering Applications of Artificial Intelligence,137*, 109070. https://doi.org/10.1016/j.engappai.2024.109070

Liao, Y., Latty, R., Genssler, P. R., Amrouch, H., & Yang, B. (2022). Wafer map defect classification based on the fusion of pattern and pixel information. In *Proceedings of IEEE international test conference (ITC)*, pp. 1–9. https://doi.org/10.1109/ITC50671.2022.00006

Liu, B., Kang, H., Li, H., Hua, G., & Vasconcelos, N. (2020). Few-shot open-set recognition using meta-learning. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. pp. 8795–8804. https://doi.org/10.1109/CVPR42600.2020.00882

Liu, C.-W., & Chien, C.-F. (2013). An intelligent system for wafer bin map defect diagnosis: An empirical study for semiconductor manufacturing. *Engineering Applications of Artificial Intelligence,26*(5–6), 1479–1486. https://doi.org/10.1016/j.engappai.2012.11.009

Loshchilov, I., & Hutter, F. (2017). SGDR: Stochastic gradient descent with warm restarts. In *International conference on learning representations (ICLR)*. pp. 1–16. https://openreview.net/forum?id=Skq89Scxx

Luo, W., & Wang, H. (2023). Composite wafer defect recognition framework based on multiview dynamic feature enhancement with class-specific classifier. *IEEE Transactions on Instrumentation and Measurement,72*, 1–12. https://doi.org/10.1109/TIM.2023.3261924

Nakazawa, T., & Kulkarni, D. V. (2018). Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing,31*(2), 309–314. https://doi.org/10.1109/TSM.2018.2795466

Nakazawa, T., & Kulkarni, D. V. (2019). Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing,32*(2), 250–256. https://doi.org/10.1109/TSM.2019.2897690

Narayan, S., Gupta, A., Khan, F. S., Snoek, C. G. M., & Shao, L. (2020). Latent embedding feedback and discriminative features for zero-shot classification. In *Proceedings of European conference on computer vision (ECCV)*. pp. 479–495. https://doi.org/10.1007/978-3-030-58542-6_29

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science,290*(5500), 2323–2326. https://doi.org/10.1126/science.290.5500.2323

Shim, J., Kang, S., & Cho, S. (2021). Active cluster annotation for wafer map pattern classification in semiconductor manufacturing. *Expert Systems with Applications,183*, 115429. https://doi.org/10.1016/j.eswa.2021.115429

Song, J., Shen, C., Yang, Y., Liu, Y., & Song, M. (2018). Transductive unbiased embedding for zero-shot learning. In *Proceeding of IEEE/CVF conference on computer vision and pattern recognition*. pp. 1024–1033. https://doi.org/10.1109/CVPR.2018.00113

Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science,290*(5500), 2319–2323. https://doi.org/10.1126/science.290.5500.2319

Wan, Z., Chen, D., & Liao, J. (2021). Visual structure constraint for transductive zero-shot learning in the wild. *International Journal of Computer Vision,129*(6), 1893–1909. https://doi.org/10.1007/s11263-021-01451-1

Wang, Z., Hao, Y., Mu, T., Li, O., Wang, S., & He, X. (2023). Bi-directional distribution alignment for transductive zero-shot learning. In *Proceedings of IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 19893–19902. https://doi.org/10.1109/CVPR52729.2023.01905

Wang, R., & Chen, N. (2019). Wafer map defect pattern recognition using rotation-invariant features. *IEEE Transactions on Semiconductor Manufacturing,32*(4), 596–604. https://doi.org/10.1109/TSM.2019.2944181

Wang, Y., Ni, D., Huang, Z., & Chen, P. (2024). A self-supervised learning framework based on masked autoencoder for complex wafer bin map classification. *Expert Systems with Applications,249*, 123601. https://doi.org/10.1016/j.eswa.2024.123601

Wang, J., Xu, C., Yang, Z., Zhang, J., & Li, X. (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

Wu, M.-J., Jang, J.-S.R., & Chen, J.-L. (2015). Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Transactions on Semiconductor Manufacturing,28*(1), 1–12. https://doi.org/10.1109/TSM.2014.2364237

Xian, Y., Lorenz, T., Schiele, B., & Akata, Z. (2018). Feature generating networks for zero-shot learning. In *2018 IEEE/CVF conference on computer vision and pattern recognition*. pp. 5542–5551. https://doi.org/10.1109/CVPR.2018.00581

Xie, G., Yang, K., Xu, C., Li, R., & Hu, S. (2022). Digital twinning based adaptive development environment for automotive cyber-physical systems. *IEEE Transactions on Industrial Informatics,18*(2), 1387–1396. https://doi.org/10.1109/TII.2021.3064364

Xu, W., Xian, Y., Wang, J., Schiele, B., & Akata, Z. (2022). Attribute prototype network for any-shot learning. *International Journal of Computer Vision,130*(7), 1735–1753. https://doi.org/10.1007/s11263-022-01613-9

Xu, Q., Yu, N., & Yu, H. (2024). Unsupervised representation learning for large-scale wafer maps in micro-electronic manufacturing. *IEEE Transactions on Consumer Electronics,70*(1), 1226–1235. https://doi.org/10.1109/TCE.2023.3262290

Yan, J., Sheng, Y., & Piao, M. (2023). Semantic segmentation-based wafer map mixed-type defect pattern recognition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,42*(11), 4065–4074. https://doi.org/10.1109/TCAD.2023.3274958

Yi, Z., Shang, W., Wang, D., Ying, M., Chen, J., Feng, W., & Wu, X. (2024). Large-margin extreme learning machines with hybrid features for wafer map defect recognition. *IEEE Transactions on Instrumentation and Measurement,73*, 1–10. https://doi.org/10.1109/TIM.2024.3374295

Yue, Z., Wang, T., Sun, Q., Hua, X.-S., & Zhang, H. (2021). Counterfactual zero-shot and open-set visual recognition. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. pp. 15399–15409. https://doi.org/10.1109/CVPR46437.2021.01515

Zhang, Y., Huang, S., Yang, W., Tang, W., Zhang, X., & Yang, D. (2023). Anchor-based discriminative dual distribution calibration for transductive zero-shot learning. *Image and Vision Computing,137*, 104772. https://doi.org/10.1016/j.imavis.2023.104772

Zhao, Z., Wang, J., Tao, Q., Li, A., & Chen, Y. (2024). An unknown wafer surface defect detection approach based on incremental learning for reliability analysis. *Reliability Engineering & System Safety,244*, 109966. https://doi.org/10.1016/j.ress.2024.109966