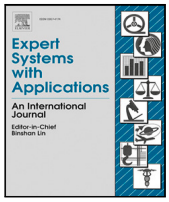
Expert Systems With Applications 249 (2024) 123601

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



A self-supervised learning framework based on masked autoencoder for complex wafer bin map classification

Yi Wang [a], Dong Ni [a,*], Zhenyu Huang [b], Puyang Chen [b]

[a] College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China
[b] Intel Corporation, Dalian 116630, China

ARTICLE INFO

Keywords:
Self-supervised learning
Masked autoencoder
Complex wafer bin map
Automatic defect classification
Semiconductor manufacturing

ABSTRACT

Wafer bin map (WBM) automatic classification is one of the critical challenges for semiconductor intelligent manufacturing. Many deep learning-based classification models have performed well in WBM classification, but all require a large amount of labeled data for training. Since real-world WBMs are highly complex and can be labeled correctly only by seasoned engineers, such requirements undermine the practical value of those methods. Several self-supervised learning methods have recently been proposed for WBM to improve classification performance. However, they still require much labeled data for fine-tuning and are only adapted for binary WBM with a single gross failure area. To address these limitations, this study introduces a self-supervised framework based on masked autoencoder (MAE) for complex WBMs with mixed bin signatures and multiple gross failure area patterns. A patchMC encoder is proposed to improve MAE's representation ability for complex WBMs with mixed bin signatures. Moreover, the pre-trained MAE encoder with a multi-label classifier fine-tuned by labeled WBMs enables a few-shot classification of complex WBMs with multiple gross failure areas. Experimental validation of the proposed method is performed on a real-world complex WBM dataset from Intel Corporation. The results demonstrate that the proposed method can make good use of unlabeled WBMs and reduce the demand for labeled data to a few-shot level and, at the same time, guarantees a classification accuracy of more than 90%. By comparing MAE with other self-supervised learning methods, MAE outperforms other existing self-supervised methods for WBM data.

1. Introduction

As the complexity of semiconductor manufacturing processes escalates, the defects in wafers generated during production have become more diverse and intricate. A wafer bin map (WBM) displays the test results for each chip on a wafer, based on the chip probe test failure mode and the chip's position (die). The chip probe test, a crucial final evaluation after the entire manufacturing process, assesses the performance and functionality of each chip. During this test, each die undergoes multiple probe test modes, with the first failure mode being recorded as the bin result.

Throughout the wafer fabrication process, various manufacturing issues may cause multiple dies on a wafer to be defective. These defects often cluster in one or several areas on the wafer, forming spatial patterns known as gross failure areas (GFAs). These patterns, including common types like ring, scratch, loc, and centers, are indicative of specific process-related issues and contain valuable data for improving yield and quality (Kim & Kang, 2021). The classification of GFAs is instrumental for engineers to identify and rectify problems in the production process, thereby reducing costs and enhancing yield (Hsu & Chien, 2007). With the growing intricacy of production environments, the need for automated WBM GFA classification has become increasingly critical.

Complex WBMs are those containing multiple gross failure areas (GFAs) and diverse bin categories, typically visualized using various colors. Fig. 1 illustrates an example of a complex WBM, detailing both the bin categories and the GFAs present. Furthermore, Fig. 2 showcases additional real-world complex WBMs, each featuring different types of GFAs and assorted bin mixtures. In practical settings, complex WBMs are most prevalent. However, for simplification, early research primarily concentrated on binary WBMs, which are derived from multi-bin WBMs based on the passing or failing bin categories (Alawieh, Boning, & Pan, 2020; Nakazawa & Kulkarni, 2019; Saqlain, Jargalsaikhan, & Lee, 2019; Wang & Ni, 2019; Wu, Jang, & Chen, 2015; Yu & Lu, 2016). In recent years, there has been a notable shift toward more nuanced classifications. Several studies have been dedicated to multi-GFA binary WBM classification (Chiu & Chen, 2021; Kim, Lee, & Kim, 2018; Kong

* Corresponding author.
E-mail addresses: 11732019@zju.edu.cn (Y. Wang), dni@zju.edu.cn (D. Ni), zhenyu.huang@intel.com (Z. Huang), puyang.chen@intel.com (P. Chen).

https://doi.org/10.1016/j.eswa.2024.123601
Received 1 September 2023; Received in revised form 28 February 2024; Accepted 1 March 2024
Available online 4 March 2024

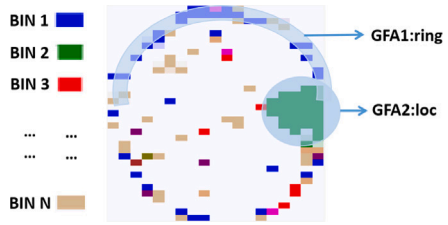


Fig. 1. Presentation of bin and GFA of an example complex WBM.

& Ni, 2020a) and single GFA multi-bin WBMs (Chen et al., 2020; Kim, Mo, Park, Kim, & Kang, 2019; Li & Tsai, 2020; Wang & Ni, 2019). Most recently, the first deep learning-based method specifically tailored for complex WBMs was introduced (Wang and Ni, 2023).

Deep learning methods typically depend on extensive labeled datasets for model training, a requirement that becomes even more pronounced with the intricate nature of complex WBMs. The sophistication of these maps substantially escalates the complexity of model structures, necessitating substantial labeled data to attain satisfactory results (Sun, Shrivastava, Singh, & Gupta, 2017). In the real-world scenario of wafer fabrication, the availability of labeled data is considerably limited due to the laborious and time-intensive nature of manual labeling. This challenge is particularly acute with WBM data, where numerous GFA categories exist, each with only a few samples. Furthermore, a significant portion of WBMs remains unlabeled. Consequently, effectively leveraging the vast amount of unlabeled data to enhance the model's classification performance with minimal labeled data is not only crucial but also profoundly significant for the semiconductor manufacturing industry.

In response to the aforementioned requirements, this study introduces a self-supervised learning framework utilizing a masked autoencoder (MAE) for complex WBM classification. MAE, recognized as an effective self-supervised method (He et al., 2021), employs vision transformers (ViT) as both encoder and decoder, marking its first application to WBMs. Employing MAE for self-supervised learning substantially enhances WBM classification by leveraging large-scale unlabeled data for pre-training, coupled with a modest amount of labeled data for fine-tuning. Initially, we present a patchMC encoder specifically designed for the multi-bin nature of complex WBMs. This encoder utilizes a convolutional neural network (CNN) preceding the MAE to bolster feature extraction capabilities for intricate WBMs. Additionally, to address the multi-GFA challenge, a multi-label fine-tuning method is implemented, facilitating both single- and multi-GFA classification. The efficacy of the proposed method is substantiated using real-world complex WBM datasets. Our experiments reveal that this method can attain over 96% classification precision with a limited number of labeled samples for complex WBMs. Furthermore, we benchmarked MAE against other prevailing self-supervised methods. The comparative results affirm that employing MAE for WBM self-supervised learning substantially ameliorates performance while concurrently curbing the reliance on labeled data.

The contributions of this article are summarized as follows:

1. A masked autoencoder (MAE) is introduced for WBM self-supervised learning, facilitating its application in classification tasks. This framework effectively harnesses large-scale unlabeled WBMs, substantially reducing the dependency on labeled data.

2. A convolutional neural network (CNN) is integrated preceding the ViT encoder in the MAE. This combination markedly enhances the MAE's feature extraction capability and significantly elevates the classification accuracy of complex WBMs.

3. The MAE encoder, pre-trained and then fine-tuned with a multi-label classifier using a small subset of labeled WBMs, ensures reliable classification performance for multi-GFA WBMs.

4. The proposed method undergoes experimental validation on a real-world complex WBM dataset provided by Intel, achieving a remarkable 96.7% classification precision on this intricate dataset with minimal labeled data.

5. Comparative analysis of MAE with other contemporary self-supervised learning methods demonstrates its superior performance for WBM data.

The remainder of this paper is structured as follows: Section 2 introduces the related work in unsupervised and self-supervised learning. Section 3 delineates the proposed framework and its components in detail. Section 4 discusses the experimental results, and finally, Section 5 concludes the paper and outlines directions for future work.

2. Related work

Over recent decades, numerous unsupervised learning methods have emerged, leveraging large-scale unlabeled data for feature representation learning (Ciresan, Meier, Gambardella, & Schmidhuber, 2010; Donoho & Grimes, 2003; Erhan et al., 2010; Goodfellow, Courville, & Bengio, 2011; Olshausen & Field, 1996; Ranzato, Boureau, & LeCun, 2007; Rifai, Vincent, Muller, Glorot, & Bengio, 2011; Roweis, 1997; Roweis & Saul, 2000; Seide, Li, & Yu, 2011; Vincent, Larochelle, Bengio, & Manzagol, 2008; Weinberger & Saul, 2004). Additionally, semi-supervised learning combines a small subset of labeled data with a vast array of unlabeled data. In the context of WBM classification, Kong and Ni (2020b) advocated a semi-supervised framework to efficiently utilize unlabeled data. More recently, self-supervised learning, a subset of unsupervised learning, has demonstrated superior performance in learning from extensive unlabeled datasets. These methods typically define a pretext task, use an encoder to learn visual features, and then transfer the learned features to downstream tasks like classification through fine-tuning with limited labeled data (Jing & Tian, 2021). Current self-supervised approaches for image visual features fall into four categories: generation-based (Goodfellow et al., 2014; Ledig et al., 2016; Pathak, Krahenbuhl, Donahue, Darrell, & Efros, 2016; Zhang, Isola, & Efros, 2016; Zhu, Park, Isola, & Efros, 2017), context-based (Caron, Bojanowski, Joulin, & Douze, 2018; Doersch, Gupta, & Efros, 2015; Jing & Tian, 2018; Li et al., 2016; Noroozi & Favaro, 2016; Noroozi, Vinjimoor, Favaro, & Pirsiavash, 2018), contrastive-based (Caron et al., 2020; Chen, Xie, & He, 2021; Grill, Strub, Altche, Tallec, & Richemond, 2020; He, Fan, Wu, Xie, & Girshick, 2020; van den Oord, Li, & Vinyals, 2018; Wu, Xiong, Yu, & Lin, 2018), and reconstruction-based (e.g., masked autoencoder) (He et al., 2021).

In the WBM field, several self-supervised learning methods have emerged. Geng, Yang, Zeng, and Yu (2021), Kahng and Kim (2021), and Wang, Ni and Huang (2023) have employed contrastive learning for WBM classification, utilizing unlabeled data to enhance feature

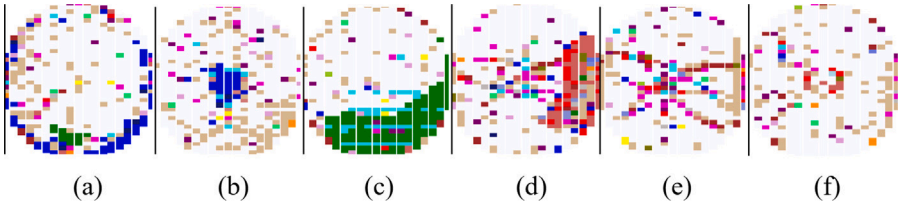


Fig. 2. Examples of more complex WBMs with different GFA types.

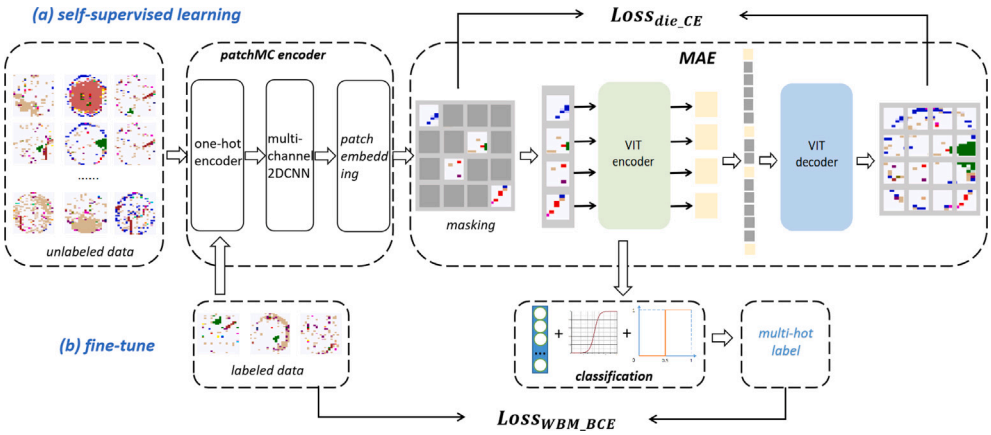


Fig. 3. The proposed method consisting of two separate steps: a self-supervised learning step and a fine-tuning step.

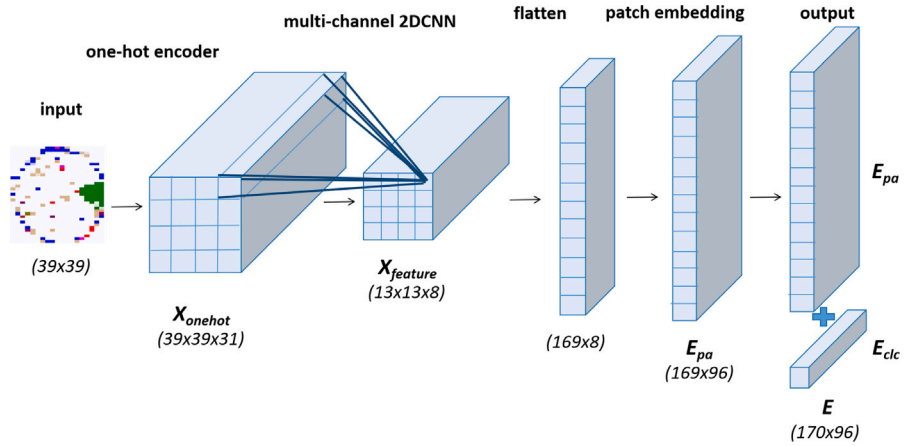


Fig. 4. The diagram of patchMC encoder.

representation learning and, subsequently, classification accuracy. Despite these advancements, current self-supervised learning approaches have notable limitations. Primarily, they do not substantially diminish the need for labeled data. During the fine-tuning phase, a significant amount of labeled data is still required to attain high classification performance. As noted in Dosovitskiy et al. (2021) and Vaswani et al. (2017), classification performance deteriorates markedly with reduced labeled data, which constrains the practicality of these methods. Additionally, existing strategies primarily address single GFA binary WBMs, limiting their applicability as most real-world WBMs are complex, encompassing multiple bin categories and GFAs. To overcome these challenges, this study introduces a self-supervised learning framework tailored for complex WBMs, requiring only minimal annotations.

3. Proposed method

3.1. Overall framework

The proposed framework for complex WBM self-supervised learning and classification is depicted in Fig. 3. It consists of two primary stages: self-supervised learning and fine-tuning. For complex WBMs, the self-supervised learning stage includes a patchMC encoder block and an MAE block.

During self-supervision, large-scale unlabeled WBMs are utilized for visual representation learning. Initially, the patchMC encoder encodes complex WBMs into feature embeddings, producing embeddings for each patch. Subsequently, the MAE engages in masking and reconstructing blocks to further enhance WBM representations using a ViT encoder. The masking block obscures a portion of the patches, while the ViT encoder transforms the remaining unmasked patches into final embeddings. The ViT decoder then reconstructs the masked patches. Post self-supervised learning, the patchMC and ViT encoder are fine-tuned with a small labeled dataset for classification. The subsequent sections detail each component.

3.2. PatchMC encoder block

The patchMC encoder, designed for complex WBMs with multiple bin categories, transforms a two-dimensional matrix representing a chip's failure test mode into a series of feature maps. These maps are then converted into randomly masked sequences incorporating one-hot encoding, multi-channel 2DCNN, patch embedding, and position embedding. One-hot encoding and multi-channel 2DCNN are pivotal innovations of this process. The encoder's diagram is shown in Fig. 4.

In a complex WBM, each matrix element signifies a bin category, a categorical variable without numerical meaning. We employ one-hot encoding to transform each bin category into a vector, converting the WBM into multi-channel 2D data (Eq. (1)):

$$X_{onehot} = Onehot(X); X \in R^{H\times W}, X_{onehot} \in R^{H\times W\times C} \quad (1)$$

Here, X represents the original WBM of size $H \times W$, and X_{onehot} denotes the transformed multi-channel 2D data, with C being the total number of bin categories.

For multi-channel WBMs, the x- and y-axes indicate spatial dimensions, while the z-axis represents the bin dimension. Complex WBMs exhibit distinct spatial features and bin mixtures across different GFAs.

The coupling of spatial and bin features is vital for identifying complex WBM GFAs. Therefore, this study uses multi-channel 2DCNN for feature extraction before MAE, as its convolution filters can extract these coupled features. The convolution filter's x- and y-axes extract spatial features, while the z-axis focuses on bin features. Adding one-hot encoding and multi-channel 2DCNN has significantly enhanced classification performance, as detailed in Section 4.2.

For the multi-channel 2D CNN layer, we employ a kernel of size (3,3,31,8) to transform the input data into feature maps (Eq. (2)):

$$X_{feature} = Conv(X_{onehot}); X_{feature} \in R^{\frac{H}{P}\times\frac{W}{P}\times f} \tag{2}$$

In this equation, $X_{feature}$ is the CNN output, P is the patch size (also the stride), and f is the filter number. Optimal values for P and f were determined to be three and eight, respectively, as shown in the experimental section.

Subsequently, these feature maps are converted into a sequence of flattened patches and mapped to embeddings E_{pa} (Eq. (3)), utilizing a network architecture comprising a linear layer of dimensions (8,96):

$$E_{pa} = Emb(X_{feature}); E_{pa} \in R^{\frac{H\times W}{P^2}\times D} \tag{3}$$

D in this equation represents the patch embedding dimension. Each patch encompasses coupled features of bin categories and spatial distributions. For optimal performance, values for P and D are set to three and 96, respectively, as discussed in the experiment section.

Following the approach in Dosovitskiy et al. (2021), a learnable class embedding E_{clc} is appended to the sequence of patch embeddings, and positional embeddings E_{pos} are integrated by addition to the retain spatial information:

$$E = concat(E_{pa}, E_{clc}) + E_{pos}; X_{clc} \in R^{1\times D}, E_{pos} \in R^{(\frac{H\times W}{P^2}+1)\times D} \tag{4}$$

The patch embeddings E serve as the input for the subsequent MAE block.

3.3. MAE block

The MAE comprises the masking block, ViT encoder, and ViT decoder. In the masking block, one part of the patches is randomly sampled as masked patches, and the remaining patches are unmasked. In this study, similar to He et al. (2021), 75% of the patch embeddings (tokens) are randomly masked, and the remaining 25% are input to the ViT encoder. Namely, the analysis has shown that different masking ratios have little effect on the results. Therefore, the masking ratio is set to the default value of 0.25. The masking formula is as follows:

$$E_{mask} = mask(E); E_{mask} \in R^{(ratio\times(\frac{H\times W}{P^2}+1))\times D} \tag{5}$$

Then, the ViT encoder processes only visible, unmasked patches. The backbone of the ViT encoder is multi-head self-attention, and the depth was set to eight; the head number was assigned to three; the multi-layer perceptron dimension was set to 512, whereas the embedding dimension was 96. Hyperparameter optimization is described in the experiment part. The encoder maps the visible patches to a latent representation that uses eight multi-headed self-attention (MSA) layers, each containing three heads. More details about these processes can be found in Vaswani et al. (2017), the paper on self-attention. The expression of the basic MSA module is as follows:

$$e'_l = MSA(LN(e_{l-1})) + e_{l-1}, l = 1...L \tag{6}$$

$$e_l = MLP(LN(e'_l)) + e'_l, l = 1...L \tag{7}$$

$$y = LN(e^0{}_L) \tag{8}$$

The MAE decoder is used only in the self-supervised learning step for masked patch reconstruction. The input data of the MAE decoder includes a complete set of tokens consisting of encoded visible patches and masked tokens (He et al., 2021), and the output data of the decoder is the reconstructed WBMs. The backbone of the ViT decoder is multi-head self-attention. The depth was set to one; the head number was assigned to eight; the multi-layer perceptron dimension was set to 64, whereas the embedding dimension was 256. In this study, The decoder reconstructs the input complex WBM by predicting the bin categories for each masked die. The bin categories differ from image pixels in the original MAE because the bin category is a categorical variable. The reconstruction of each bin is analogous to a die-level classification.

Therefore, the decoder's architecture and reconstruction target is changed. After the ViT of the decoder, a linear projection layer is added to the last layer to map the decoder's output embedding to a C-dimension embedding, where C equals the total number of bin categories. Then, a softmax layer is added to the end of the linear projection layer to obtain the predicted bin category of each die. The softmax activation function is defined as follows:

$$p_c(x) = \exp(a_c(x))/(\sum_{c'}^{C}\exp(a_{c'}(x))) \tag{9}$$

where $a_c(x)$ denotes the activation function in the bin dimension c at the die position x, where $x \in X_{output}$; $p_c(x)$ is the approximated maximum-function.

The baseline MAE method is used for images with RGB pixels, and the reconstruction process is based on pixel-level prediction. Since an RGB is a continuous numerical value, the mean square error (MSE) function is selected as the loss function (He et al., 2021). In this study, the element is the bin category of each die. Thus, we proposed to use a die-level cross-entropy loss to compute the cross-entropy between the reconstructed and original WBMs in the die-level on masked dice instead of the mean square error (MSE) loss in the baseline MAE. The loss function is given by:

$$Loss_{die\_CE} = \sum_{x=1}^{N}\sum_{i=1}^{C}\omega_{(x)}(i)\log p_{(x)}(i) \tag{10}$$

where $\omega_{(x)}$ is the one-hot encoding of the true bin category of the xth die and $p_{(x)}(i)$ denotes the ith bin prediction probability of the xth die. N equals the total number of die on a wafer bin map and C equals the total number of bin category.

3.4. Multi-label fine-tuning block

A multi-GFA WBM has multiple GFA types, as shown in Fig. 3(b). Typically, after the self-supervised pre-training phase, a dataset labeled for a single class per instance is employed for the downstream classification task. The loss function used in this scenario is cross-entropy, and the activation function in the final layer is softmax, referred to as the "single-label fine-tuning" method in this study. However, employing single-label fine-tuning for multi-GFA WBMs might significantly affect the classification performance, leading to a decrease in overall accuracy. To address the multi-GFA classification problem, this study applies a multi-label fine-tuning method. The multi-label classifier consists of a patchMC encoder, a ViT encoder, and a classification head with a linear layer and an activation function. The patchMC and ViT encoders' initialization weights are transferred from the MAE self-supervised pre-training, while the classification head's weights are randomly initialized. The ViT decoder of MAE is not utilized during the fine-tuning period.

The multi-label fine-tuning differs from single-label fine-tuning in three ways. First, the ground truth is a multi-hot vector, compared to the one-hot vector in single-label fine-tuning. In a multi-hot vector, ones indicate existing GFA types, while zeros represent non-existent ones. Second, the activation function is sigmoid instead of softmax, mapping each value in an n-dimensional vector to the range of (0,1), allowing for independent values unlike softmax. This function enables

the simultaneous assessment of multiple classes' existence. The sigmoid function is defined as (Eq. (11)):

$$p_n(x) = \frac{1}{1+\exp(-a_n(x))} \tag{11}$$

Here, $a_n(x)$ is the activation in the class dimension n, and $p_n(x)$ is the predicted probability vector. After applying the sigmoid function, each predicted value represents the likelihood of a particular GFA type's existence. To determine if a GFA exists, each predicted value is compared with a threshold. If the value exceeds 0.5, the predicted label for that GFA is set to one, indicating its presence:

$$L_n(x) = \begin{cases} 1 & , p_n(x) \geq 0.5 \\ 0 & , p_n(x) < 0.5 \end{cases} \tag{12}$$

Third, the loss function employs binary cross-entropy (BCE) instead of cross-entropy, adding a penalty for non-1 predictions and maximizing the value at the "1" position while minimizing others. BCE is suitable for binary targets (0 or 1) and is defined as (Eq. (13)):

$$Loss_{BCE} = -\frac{1}{n}\sum_{x=1}^{n}(w(x)\log(p(x)) + (1-w(x))\log(1-p(x))) \tag{13}$$

Here, n is the total number of classes, $w(x)$ is the true label of xth class (0 or 1) and $p(x)$ is the predicted probability that belong to class x.

In the inference phase, the multi-label classifier maps each input WBM to an n-dimensional predicted vector. The patchMC and MAE encoders transform a WBM into a feature embedding, and the classifier outputs the predicted vector. Each value is compared with a preset threshold to determine the classification results.

4. Experimental results

4.1. Data and experimental setup

In the experiments, we utilized a real-world complex WBM dataset collected by Intel to verify the performance of the proposed framework and conduct an ablation study on the patchMC encoder and multi-label fine-tuning. The complex WBM dataset comprises 180,000 unlabeled WBMs for self-supervised learning and 2070 labeled WBMs spanning 12 GFA types for the fine-tuning period. For IP protection, this paper does not disclose the categories and examples of the 12 GFAs in the dataset. Many WBMs in the dataset feature multiple GFA types labeled by experienced engineers. The 2070 labeled dataset represented a mix of multi- and single-label WBMs. To assess the classification performance with small-scale labeled data for fine-tuning, we randomly sampled 240 labeled data (20 to 40 for each GFA) from all labeled datasets as training data for the fine-tuning period, with the remaining labeled data serving as test data. The total number of labels for each GFA is 31, 27, 28, 29, 22, 41, 31, 24, 25, 20, 20, 20, respectively. We acknowledge that 20 to 40 labels are a manageable amount for manual labeling by engineers in practice. Additionally, we sampled two few-shot trainsets to evaluate the few-shot classification performance of the proposed method, with details presented later. A complex WBM is a 39 × 26 2D matrix. We resized the original complex WBM to 39 × 39 square matrices, and the total number of bin categories is 31. Thus, after one-hot encoding, a complex WBM transformed into multi-channel 2D data with the size of 39 × 39 × 31, with each channel indicating one bin category. The patch size was set to three, corresponding to 3 × 3 dies.

The hyperparameters of the patchMC and ViT encoders are utilized in both self-supervised learning and fine-tuning periods. Thus, we optimized them by comparing classification performance after fine-tuning with varying hyperparameters. For the patchMC encoder, we compared different CNN depths and widths following Wang, Ni (2023). The results indicate that two convolution layers with 8 and 16 filters and a 3 × 3 filter size lead to better classification performance. For the ViT encoder, we compared classification performance with various multi-layer perceptron dimensions, embedding dimensions, head numbers, and the number of multi-head self-attention layers. The results suggest optimal performance with $mlp = 512$, $dim = 96$, $head = 3$, and $depth = 8$.

In the self-supervised pre-training process, the batch size was 2048; the optimizer was Adam, and the learning rate was 3e−4. The parameter selection was based on experimental results, optimizing the model's classification performance. The self-supervised method was implemented using Python 3.8.2 and Pytorch 1.9.0 on four Nvidia V100 GPUs with 16 GB memory each.

The patchMC and ViT encoders were initialized for the fine-tuning period with weights learned during self-supervised pre-training. The classification block parameters were randomly initialized. The output vector's dimension equals 12, the total number of GFA types. In the fine-tuning process, the batch size was 64, the optimizer was Adam, and the learning rate was 3e−4. The classification precision, recall, and F1-score were selected as evaluation indices and calculated after fine-tuning to assess both self-supervised learning and fine-tuning performance. The definitions of precision, recall, and F1-score are as follows:

$$Precision = \frac{TP}{TP+FP} \tag{14}$$

$$\mathrm{Re}call = \frac{TP}{TP+FN} \tag{15}$$

$$F1-score = \frac{2 * \mathrm{Pr}ecision * \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \tag{16}$$

In these equations, TP denotes the number of positive samples correctly identified; FP is the number of negative samples incorrectly labeled as positive; FN is the number of positive samples incorrectly labeled as negative.

To confirm the advantages of MAE over existing WBM self-supervised learning methods, we compared MAE with two representative existing works. For a fair comparison, we used the same WM-811K dataset as the existing methods. WM-811K contains only single-label binary WBMs. Examples of nine GFA types in WM-811K are illustrated in Fig. 5. Further experiment details are provided in Section 4.5. Furthermore, mixedWM38 is a publicly available binary WBM dataset characterized by mixed-type defect patterns. To evaluate the classification performance of our proposed method on binary WBMs with multi-label capabilities, we conducted additional experiments on this dataset. The specific details of these experiments are elaborated in Section 4.6.

4.2. Ablation study for patchMC encoder on complex WBM

To assess the effectiveness of the proposed patchMC encoder for complex WBMs, we compared two self-supervised learning scenarios: MAE with the patchMC encoder and MAE without it. Initially, the unlabeled complex WBM dataset was used for self-supervised learning. Subsequently, the labeled training set was employed for multi-label fine-tuning, and the test set was utilized to calculate classification precision, reflecting the performance of self-supervised learning. When MAE is without the patchMC encoder, a complex WBM is transformed into a three-channel, 64 × 64 pixel RGB image, and the complex WBMs are input directly into the ViT encoder of MAE without one-hot encoding and multi-channel 2DCNN.

The overall classification precision of the test set for the two cases are 0.956 and 0.558, respectively. These results demonstrate that the patchMC encoder significantly enhances self-supervised learning performance for complex WBMs, thereby improving MAE's representation capability. This improvement is primarily due to one-hot encoding, which effectively expresses bin information by transforming each bin category into a vector, and the multi-channel 2DCNN, which extracts bin and spatial dimension coupling features using convolution filters.

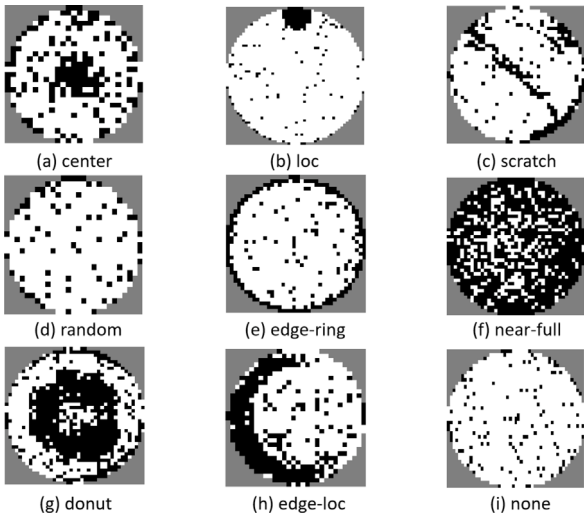


Fig. 5. Examples of nine GFA types in WM-811K.

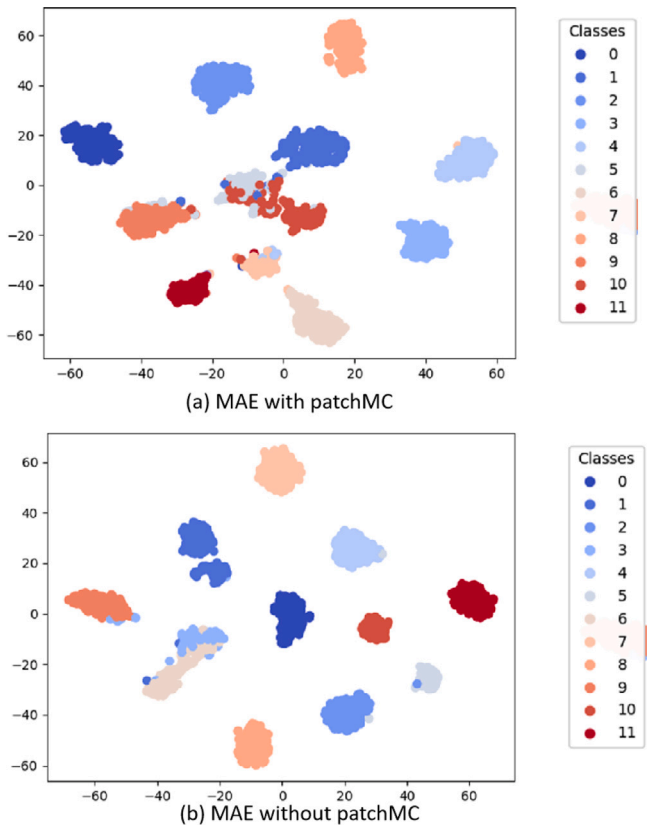


Fig. 6. Visualization of 12 GFA types representation with and without patchMC encoder using t-SNE.

Such feature extraction is crucial for complex WBM as it consists of multiple bin categories with core information residing in the coupling of bin and spatial features. Without the patchMC encoder, classification accuracy drops considerably as categorical bins are treated as numerical RGB values, and the coupling features between bin and spatial dimensions cannot be effectively extracted.

To elucidate the differences between MAE with and without the patchMC encoder for each GFA type, post fine-tuning, the embedding distribution of the 12 GFA types after the ViT encoder of MAE is visualized using t-SNE, a technique for dimensionality reduction (van der Maaten & Hinton, 2008). As depicted in Fig. 6, with the patchMC encoder, the 12 GFA types are well-separated, enabling precise classification. Without the patchMC encoder, there is confusion between classes 5 and 10, with many samples incorrectly classified as classes 7 and 9. These results confirm that the patchMC encoder enhances MAE's feature representation ability for GFAs that are challenging to distinguish.

Furthermore, to evaluate the impact of one-hot encoding within the patchMC encoder, we compared performance between patchMC embedding on raw bin values (without one-hot) and one-hot encoding vectors (with one-hot). The macro F1-scores for these methods are 0.929 and 0.974, respectively, indicating a 4.5 percent improvement with one-hot encoding. Notably, for GFA4 and GFA7, the F1 scores improved by 11.7 percent and 15.4 percent, respectively.

4.3. Comparison with existing methods on complex WBMs

To verify the advantages of the proposed method over other existing self-supervised learning approaches for complex WBM datasets, we compared it with several established self-supervised algorithms and CNN-based supervised methods, respectively.

For comparing with existing self-supervised methods on complex WBMs, we selected two highly representative and well-known methods for comparison: momentum contrastive learning (MoCo) (He et al., 2020) and BYOL (Grill et al., 2020), both utilizing the CNN baseline model. For MoCo and BYOL, random rotation was applied during the data augmentation period. In the fine-tuning period, all comparison methods employed the multi-label classification strategy introduced in Section 3.4. The same with the proposed method, 180,000 unlabeled WBMs were used for MoCo and BYOL pre-training, and the labeled training and test sets were used for the fine-tuning period.

The comparison results with existing methods are shown in Table 1. The results indicate that our proposed MAE-based self-supervised method surpasses the performance of these established self-supervised learning approaches, further substantiating the efficacy and potential of our methodology.

For comparison with CNN-based supervised methods, a previously proposed CNN-based classification method for complex WBMs (Wang, Ni, 2023) consists of two 3×3 convolution layers with eight and 16 filters, respectively, with stride and padding set to one. Additionally, two 2×2 max-pooling layers follow the convolution layers. We first compared the proposed method with this CNN baseline. Furthermore, given that a robust data augmentation strategy can significantly enhance classification performance, especially for limited and complex real-world labeled WBM datasets, we also implemented CNN with data augmentation for comparison. We adopted four standard data augmentation techniques for WBMs: rotation, flip, crop, and denoising. For the augmented CNN scenario, the labeled training dataset containing 240 samples was expanded to 1680 labeled samples, and the CNN model was trained on this augmented dataset. Meanwhile, the non-augmented CNN model was trained using only the initial 240 labeled samples. During the testing phase, the same test set was employed to assess the multi-label classification performance of both CNN models.

The comparison results with CNN baselines are presented in Table 2, demonstrating that data augmentation indeed enhances CNN classification performance. Moreover, the results confirm that our proposed method outperforms the CNN-based supervised learning approach in terms of classification effectiveness.

4.4. Few-shot classification performance evaluation for complex WBMs

To evaluate the proposed method's classification performance in few-shot scenarios, we randomly sampled 5 and 3 labeled data for each GFA from the labeled complex WBM dataset as training data for the fine-tuning period. The remaining labeled data from the complex WBM dataset were used as test data. Following self-supervised learning using

Table 1
The GFA-level classification performance comparison with existing methods on complex WBM dataset.

GFA types	MoCo (He et al., 2020)			BYOL (Grill et al., 2020)			Ours		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
1	0.852	0.795	0.823	0.880	0.943	0.910	0.995	0.991	0.993
2	1.000	0.979	0.989	0.974	0.966	0.970	1.000	1.000	1.000
3	0.989	0.895	0.940	0.939	0.840	0.887	1.000	0.950	0.974
4	0.789	0.808	0.798	0.726	0.784	0.754	0.845	0.960	0.899
5	0.990	0.975	0.983	0.995	1.000	0.998	1.000	1.000	1.000
6	0.807	0.880	0.842	0.921	0.930	0.925	1.000	0.970	0.985
7	0.920	0.752	0.828	0.815	0.626	0.708	0.979	0.825	0.845
8	0.982	1.000	0.991	0.991	0.986	0.989	1.000	1.000	1.000
9	0.942	0.942	0.942	0.972	0.904	0.937	1.000	0.995	0.997
10	0.969	0.959	0.964	0.965	1.000	0.982	0.990	0.995	0.997
11	1.000	0.953	0.976	0.981	0.972	0.977	0.991	1.000	0.995
12	0.994	0.989	0.992	0.968	0.995	0.981	1.000	1.000	1.000
Macro	0.936	0.911	0.922	0.927	0.912	0.918	**0.983**	**0.974**	**0.974**

Table 2
The GFA-level classification performance comparison with CNN baselines on complex WBM dataset.

GFA types	CNN (Wang, Ni, 2023)			CNN_aug			Ours		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
1	0.861	0.886	0.873	0.985	0.938	0.961	0.995	0.991	0.993
2	0.910	0.983	0.945	0.908	0.996	0.950	1.000	1.000	1.000
3	0.943	0.920	0.932	0.965	0.960	0.962	1.000	0.950	0.974
4	0.690	0.640	0.664	0.614	0.904	0.731	0.845	0.960	0.899
5	0.980	0.975	0.978	0.985	0.984	0.983	1.000	1.000	1.000
6	0.839	0.940	0.887	0.979	0.950	0.964	1.000	0.970	0.985
7	0.955	0.256	0.404	0.976	0.167	0.285	0.979	0.825	0.845
8	0.896	0.977	0.935	0.969	0.982	0.975	1.000	1.000	1.000
9	0.945	0.915	0.930	0.963	0.963	0.963	1.000	0.995	0.997
10	0.889	0.907	0.898	0.963	0.938	0.950	0.990	0.995	0.997
11	0.693	0.888	0.779	0.980	0.935	0.957	0.991	1.000	0.995
12	0.956	0.962	0.959	0.952	0.989	0.970	1.000	1.000	1.000
Macro	0.880	0.854	0.849	0.937	0.892	0.888	**0.983**	**0.974**	**0.974**

Table 3
The 5-shot classification performance for complex WBM (%).

GFA type	F1-score	GFA type	F1-score
1	97.6	7	79.0
2	97.6	8	100
3	96.1	9	99.5
4	84.7	10	98.7
5	98.3	11	96.1
6	96.4	12	99.7

Table 4
The 3-shot classification performance for complex WBM (%).

GFA type	F1-score	GFA type	F1-score
1	97.9	7	79.4
2	97.2	8	100
3	97.2	9	99.7
4	86.5	10	98.7
5	99.0	11	90.3
6	98.0	12	98.9

an unlabeled complex WBM dataset, 5-shot and 3-shot classifications were conducted. The GFA-level classification accuracy test results are presented in Tables 3 and 4.

The results show that both 3-shot and 5-shot classification performances are above 91%, based on MAE's self-supervised pre-training. The F1-scores for 3-shot and 5-shot are close, indicating that the complex WBM classifier, fine-tuned with only three labeled data for each GFA, can achieve over 90% classification accuracy. This finding demonstrates that our proposed self-supervised method can effectively utilize unlabeled data to learn a precise representation of complex WBM and significantly reduce the demand for labeled data to a few-shot level without additional design. The performance for GFA4 and GFA7 is relatively lower, likely because these GFAs often co-occur in a WBM, making multi-GFA classification more challenging due to mutual interference.

4.5. Comparison with existing methods on single-label binary WBM dataset

TSM'21 (Kahng & Kim, 2021) and ICCAD'21 (Geng et al., 2021) are two notable works published in recent years that focus on self-supervised learning methods for WBM classification. Consequently, we selected these works for comparison with our MAE to evaluate the performance of self-supervised learning. Both works utilize the WM-811K dataset, which contains only single-label binary WBMs. To ensure a fair comparison, we also used the same labeled and unlabeled data in our method. For the WM-811K dataset, we employed MAE for binary WBM self-supervised learning, omitting the patchMC encoder. For each comparison method, the unlabeled data in WM-811K were first used for self-supervised pre-training. Then, the classifier was fine-tuned with labeled data, and test data were used to calculate classification performance, evaluating the self-supervised learning method's effectiveness. A detailed description of the comparison methods and results are presented in the following.

The first comparison method is TSM'21 (Kahng & Kim, 2021), where self-supervision is achieved by minimizing a contrastive loss function that encourages features extracted from the original, non-augmented WBM, and its augmented counterparts to cluster together. A memory bank is utilized to mitigate the computational burden of negative sampling in the context of noise contrastive estimation (NCE). Following the work in Kahng and Kim (2021), the entire labeled data were split into training, validation, and test sets in a 0.8:0.1:0.1 ratio. The amount of labeled training data varied from a minimum of 1383 samples (1% of the total training data) to a maximum of 138,360 samples (100%). These data are used to investigate how the proposed method scales at

Table 5
Classification accuracy comparison of TSM'21 and our proposed method for self-supervison.

Methods	Labeled data					
	1%	5%	10%	25%	50%	All
Supervised TSM'21 (Kahng & Kim, 2021)	0.554	0.681	0.711	0.782	0.846	0.897
Self-supervised TSM'21 (Kahng & Kim, 2021)	0.741	0.815	0.839	0.864	0.880	0.897
Supervised ours	0.508	0.789	0.804	0.819	0.838	0.878
Self-supervised ours	**0.813**	**0.865**	**0.874**	**0.886**	**0.905**	**0.924**

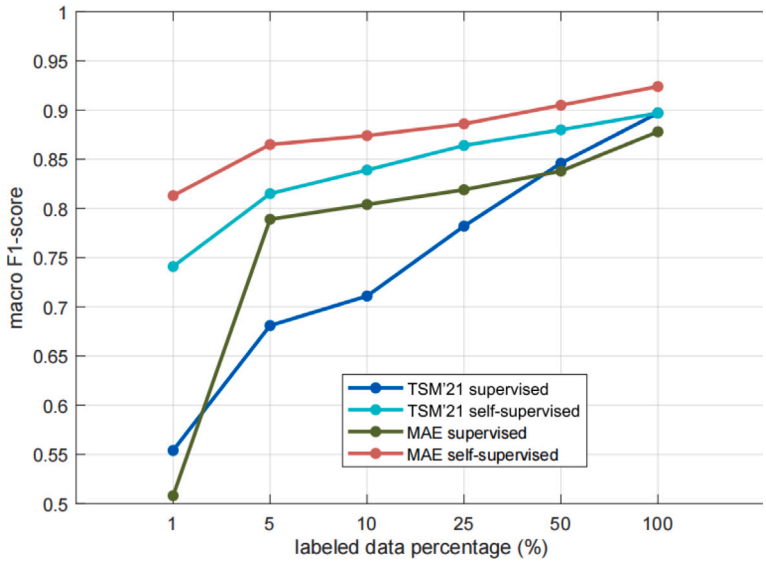


Fig. 7. The classification accuracy comparison after TSM'21 and the proposed method.

different amounts of labeled data. The results are also compared with the supervised baselines of TSM'21 and our proposed method.

The classification results of TSM'21, the proposed method, and the corresponding supervised baseline are shown in Table 5. The results indicate that the proposed method macro-averagely outperformed TSM'21 with a 7.2% improvement at 1% labeled data, 5.0% improvement at 5% labeled data, 3.5% improvement at 10% labeled data, 2.2% improvement at 25% labeled data, 2.5% improvement at 50% labeled data, and 2.7% improvement in macro F1-score at all labeled data. Thus, performance improved significantly for different labeled data scales consistently. This advantage is even more apparent with fewer labeled data samples. Additionally, the number of fine-tuning epochs for TSM'21 was 100, while that for the proposed method was only 30. This indicates that the proposed classification model could converge quickly and effectively reduce calculation time. The results of the four comparison methods at different labeled data scales are presented in Fig. 7, illustrating that MAE pre-training can learn a good representation for WBM and reduce the command for labeled data.

The second comparison method is ICCAD'21 (Geng et al., 2021), which employs an end-to-end wafer defect classifier combining few-shot learning and self-supervised learning algorithms. ICCAD'21 addresses two primary tasks: one focuses on prototypical network-based few-shot classification to mitigate the wafer data imbalance issue, and the other leverages contrastive learning-based self-supervised learning on unlabeled WBMs. Following the approach in Geng et al. (2021), 50,000 unlabeled data from WM-811K were randomly selected for unsupervised pre-training. The labeled dataset was divided into training and test sets with a 0.6:0.4 ratio. Precision, recall, and F1-score were calculated for each GFA to quantitatively assess the WBM classification performance. To evaluate the performance on the imbalanced dataset, the macro-average was used to calculate the average evaluating index of all GFA types. The comparison results of ICCAD'21 on the WM-811K dataset are presented in Table 6. These results aim to highlight the efficacy of the proposed method in handling imbalanced datasets and its superiority in self-supervised learning for WBM classification.

Table 6
Classification performance comparison of the proposed method with ICCAD' 21 for self-supervison.

GFA types	ICCAD'21 (Geng et al., 2021)			Ours		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Center	0.736	0.950	0.830	**0.911**	**0.991**	**0.984**
Donut	0.806	**0.842**	0.824	**0.909**	0.788	**0.844**
Edge-Loc	0.647	0.802	0.716	**0.971**	**0.981**	**0.976**
Edge-Ring	0.992	0.921	0.955	**1.000**	**0.999**	**0.999**
Location	0.605	0.720	0.658	**0.919**	**0.938**	**0.928**
Near-Full	0.810	0.867	0.840	**0.947**	**0.900**	**0.923**
Random	0.816	0.652	0.724	**0.988**	**0.893**	**0.938**
Scratch	0.474	**0.701**	**0.565**	**0.671**	0.593	0.629
None	0.986	0.967	0.977	**1.000**	**1.000**	**1.000**
Macro	0.764	0.825	0.788	**0.931**	**0.898**	**0.914**

The evaluation metrics, including precision, recall, and F1-score along with their corresponding macro-averaged values, were calculated for the three methods. The results reveal that the proposed method surpasses ICCAD'21, demonstrating significant improvements in the macro-averaged precision, recall, and F1-score by 16.7%, 7.3%, and 12.6% respectively. Although ICCAD'21 adopted a combination of few-shot learning and self-supervised learning to address the issue of imbalanced training datasets, our proposed self-supervised method's classification performance still substantially outperforms it. Furthermore, our method is characterized by its simplicity, speed, and efficiency, distinguishing it from other published WBM self-supervised learning methods. This makes it not only effective but also practical for real-world application scenarios where rapid and reliable classification is essential.

Moreover, we compared MAE with two of the well-known and effective self-supervised learning methods, momentum contrastive learning

Table 7
Classification performance comparison of two well-known self-supervised learning methods.

Amount of labeled data	MAE (He et al., 2021)	MoCo (He et al., 2020)	BYOL (Grill et al., 2020)
20 each class (0.1%)	**0.733**	0.725	0.718
30 each class (0.15%)	**0.764**	0.738	0.744
50 each class (0.25%)	**0.803**	0.772	0.771
100 each class (0.5%)	**0.820**	0.794	0.814
200 each class (1%)	**0.880**	0.826	0.857
All (100%)	**0.917**	0.910	0.907

(MoCo) (He et al., 2020; Wang, Ni, Huang, 2023) and BYOL (Grill et al., 2020). Referring to Wang, Ni, Huang (2023), the baseline of the comparison methods is a 2D convolution neural network (CNN) consisting of two 5 × 5 convolution layers with filter numbers of eight and 16, respectively. Two 2 × 2 max-pooling layers are added after the convolution layers. Random rotation is applied in the data augmentation period. 50,000 unlabeled data of WM-811K were randomly sampled and used for self-supervised pre-training. The labeled data were randomly split into the training and test sets according to the ratio of 0.8:0.2. Several small-scale balanced labeled datasets for eight GFA types, except the "none" type, were randomly sampled from the training set for fine-tuning. The number of training data was increased from 20 for each GFA type (0.1% of the total training set) to all of the training set (100%) of WM-811K.

The comparison results are presented in Table 7. The findings reveal that employing MAE for WBM self-supervised learning and fine-tuning with just 1% of all labeled data leads to over 80% classification accuracy, outperforming other self-supervised methods. Notably, the performance at 1% of all labeled training data reaches 88%, closely approaching the 91% obtained with the entire labeled dataset. Furthermore, the classification accuracy exceeds 73% with only 0.1% of all labeled training data. This underlines that MAE pre-training can learn an exceptional visual representation, significantly alleviating the need for labeled data.

4.6. Performance evaluation on multi-label binary WBM dataset

MixedWM38 is a publicly available binary WBM dataset characterized by mixed-type defect patterns. To evaluate the classification performance of our proposed method on binary WBMs with such complex defect patterns, we conducted experiments on this dataset. Originally published by Wang, Xu, et al. (2020), MixedWM38 includes one fault-free pattern, eight single defect patterns, thirteen 2 mixed-type patterns, twelve 3 mixed-type defect patterns, and four 4 mixed-type defects, totaling 38,015 WBMs. For WBMs with a single defect pattern, the ground truth is represented by 8-dimensional one-hot vectors. In contrast, for WBMs with multiple defect patterns, 8-dimensional multi-hot vectors are used, where ones indicate the presence of specific defect patterns, and zeros denote their absence. The dataset consists of WBMs of 52 × 52 dimensions, which we resized to 64 × 64 for our method. In the pre-training phase, all 38,015 data were utilized. As MixedWM38 contains binary WBMs, we trained the MAE directly without the patchMC encoder. After pre-training, we applied the multi-label fine-tuning method based on the pre-trained ViT encoder for classification. The data were randomly split into an 8:2 ratio for training and testing.

Table 8 presents the classification performance. Similar to Table 1, we provided the defect-level classification precision, recall, and F1-score for the eight defect patterns, illustrating the classification performance of each defect pattern within the WBMs containing them. The results show that our proposed MAE pre-training and multi-label fine-tuning method can effectively classify binary WBMs with mixed-type defects, achieving a macro precision of 0.973, a macro recall of 0.972, and a macro F1-score of 0.972. These findings demonstrate the efficacy of our method in handling binary WBMs with mixed-type defects.

Table 8
Classification performance evaluation of the proposed method on MixedWM38 dataset.

Defect pattern	Precision	Recall	F1-score
Center	0.983	0.994	0.988
Donut	1.000	0.993	0.997
Edge-Loc	0.944	0.959	0.952
Edge-Ring	0.971	0.966	0.969
Location	0.980	0.963	0.971
Near-Full	0.963	0.963	0.963
Scratch	0.954	0.942	0.947
Random	0.988	0.994	0.991
Macro	0.973	0.972	0.972

4.7. Discussion

From the experiment results for complex WBMs, employing large-scale unlabeled data for self-supervised learning with an enhanced MAE and a patchMC encoder, a multi-label classifier fine-tuned with a small set of labeled data achieves a 96.7% classification precision. Furthermore, when comparing MAE with established self-supervised learning methods and CNN-based supervised baselines, our results support that MAE outperforms these comparative methods for complex WBM data.

The few-shot classification results reveal that both 3-shot and 5-shot classification accuracies exceed 90%, indicating that as few as 3 or 5 labeled data for each GFA type are sufficient to fine-tune an effective classifier. This finding demonstrates that the proposed method can effectively learn from large-scale unlabeled WBMs, significantly reducing the reliance on labeled data without necessitating additional design. Minimizing the need for labeled data is essential, especially for real-world applications.

In comparison with existing self-supervised learning methods on single-label binary WBM dataset WM-811K, our proposed approach demonstrated substantial improvements in the quality of self-supervised learning. Notably, there was a +7.2% macro F1-score improvement for classification with only 1% of the total training labels compared to TSM'21. Additionally, our method achieved a +12.6% macro F1-score improvement compared to ICCAD' 21. Moreover, for the multi-label binary WBM dataset MixedWM38, our proposed approach also exhibited effective classification performance, achieving a macro F1-score of 0.972. These results underscore the superior efficacy and potential of our proposed methodology.

5. Conclusion

Large-scale unlabeled WBMs are available in real-world semiconductor manufacturing, and labeled WBMs are challenging to obtain. Therefore, using large-scale unlabeled data and reducing the demand for labeled data is essential for WBM analysis. Existing WBM self-supervised methods learn WBM representation from unlabeled data but still need a large amount of labeled data to fine-tune an ideal classification model. This study leverages a self-supervised learning approach based on MAE that can improve classification accuracy with only a few labeled data for fine-tuning. Complex WBMs are the most common data in wafer fabrication, containing multiple bin categories and GFA types. This study first proposes a complex WBM self-supervised learning

method. For the multi-bin problem, we proposed a patchMC encoder to improve MAE's feature representation capability for complex WBM. In addition, after self-supervised learning, a multi-label fine-tuning method is proposed for multi-GFA WBM classification.

The performance of the proposed framework is evaluated on a real-world complex WBM dataset. Ablation studies prove the necessity and effectiveness of patchMC encoder. Few-shot classification results indicate that the proposed framework can achieve a high classification accuracy with few labeled data. Moreover, by comparing MAE with other existing self-supervised learning methods, we proved that using MAE for WBM self-supervised learning outperforms other existing self-supervised learning approaches.

Finally, future work will focus on the object detection approaches combined with self-supervised learning and few-shot learning to detect more complex GFA with local features and very weak signals to improve the classification performance for complex WBMs.

CRediT authorship contribution statement

Yi Wang: Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Dong Ni:** Conceptualization, Supervision, Project administration, Writing – review & editing, Funding acquisition. **Zhenyu Huang:** Methodology, Data curation, Visualization, Resources, Investigation. **Puyang Chen:** Software, Formal analysis, Validation.

Declaration of competing 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.

Data availability

The data that has been used is confidential.

Acknowledgments

The authors would like to acknowledge the financial support from the National Natural Science Foundation of China (Grant No. 62173298).

References

Alawieh, M. B., Boning, D., & Pan, D. Z. (2020). Wafer map defect patterns classification using deep selective learning. In *ACM/IEEE design automation conference* (pp. 1–6).

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *European conference on computer vision*.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. In *Neural information processing systems (neurIPS)*.

Chen, X., Xie, S., & He, K. (2021). An empirical study of training self-supervised vision transformers. In *IEEE conference on computer vision and pattern recognition*.

Chen, L. L.-Y., et al. (2020). TestDNA-E: Wafer defect signature for pattern recognition by ensemble learning. In *IEEE international test conference*.

Chiu, M., & Chen, T. (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.

Ciresan, D., Meier, U., Gambardella, L., & Schmidhuber, J. (2010). Deep big simple neural nets for handwritten digit recognition. *Neural Computation*, *22*, 1–14.

Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *IEEE international conference on computer vision* (pp. 1422–1430).

Donoho, D., & Grimes, C. (2003). *Hessian Eigenmaps: New Locally Linear Embedding Techniques for High-Dimensional Data*: *Technical Report 2003–08*, Dept. of Statistics, Stanford Univ..

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations*. ICLR.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, *11*, 625–660.

Geng, H., Yang, F., Zeng, X., & Yu, B. (2021). When wafer failure pattern classification meets few-shot learning and self-supervised learning. In *IEEE international conference on computer aided design*.

Goodfellow, I., Courville, A., & Bengio, Y. (2011). Spike-and-slab sparse coding for unsupervised feature discovery. arXiv preprint arXiv:1201.3382.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Grill, J. B., Strub, F., Altche, F., Tallec, C., & Richemond, P. (2020). Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in neural information processing systems*.

He, K., Chen, X., Xie, S., Li, Y., Dollar, P., & Girshick, R. (2021). Masked autoencoders are scalable vision learners. In *IEEE conference on computer vision and pattern recognition*.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *IEEE conference on computer vision and pattern recognition*.

Hsu, S. C., & Chien, C. F. (2007). Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing. *International Journal of Production Economics*, *107*(1), 88–103.

Jing, L., & Tian, Y. (2018). Self-supervised spatiotemporal feature learning by video geometric transformations. In *IEEE conference on computer vision and pattern recognition*.

Jing, L., & Tian, Y. (2021). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(11), 4037–4058.

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.

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.

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

Kim, J., Mo, K., Park, J., Kim, H., & Kang, P. (2019). Bin2Vec: A better wafer bin map coloring scheme for comprehensible visualization and effective bad wafer classification. *Applied Sciences*, *597*(9).

Kong, Y., & Ni, D. (2020a). Qualitative and quantitative analysis of multi-pattern wafer bin maps. *IEEE Transactions on Semiconductor Manufacturing*, *33*(4), 578–586.

Kong, Y., & Ni, D. (2020b). A semi-supervised and incremental modeling framework for wafer map classification. *IEEE Transactions on Semiconductor Manufacturing*, *33*(1), 62–71.

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., et al. (2016). Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE conference on computer vision and pattern recognition* (pp. 4681–4690).

Li, D., Hung, W.-C., Huang, J.-B., Wang, S., Ahuja, N., & Yang, M.-H. (2016). Unsupervised visual representation learning by graph-based consistent constraints. In *European conference on computer vision*.

Li, K. S., & Tsai, N. C. (2020). TestDNA: Novel wafer defect signature for diagnosis and pattern recognition. *IEEE Transactions on Semiconductor Manufacturing*, *33*(3), 383–390.

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.

Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*.

Noroozi, M., Vinjimoor, A., Favaro, P., & Pirsiavash, H. (2018). Boosting self-supervised learning via knowledge transfer. ArXiv preprint arXiv:1805.00385.

Olshausen, B., & Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*, 607–609.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *IEEE conference on computer vision and pattern recognition* (pp. 2536–2544).

Ranzato, M., Boureau, Y., & LeCun, Y. (2007). Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*.

Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. In *Int'l conf. machine learning*.

Roweis, S. (1997). *EM Algorithms for PCA and Sensible PCA*: *Technical Report CNS-TR-97-02*, California Institute of Technology.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

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

Seide, F., Li, G., & Yu, D. (2011). Conversational speech transcription using context-dependent deep neural networks. In *Conference international speech communication association* (pp. 437–440).

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*.

van den Oord, A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv:1807.03748.

van der Maaten, L., & Hinton, G. E. (2008). Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems (neurIPS)*.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In *Int'l conf. machine learning*.

Wang, Y., & Ni, D. (2019). Multi-bin wafer maps defect patterns classification. In *International symposium on semiconductor manufacturing intelligence*.

Wang, Y., & Ni, D. (2023). A deep learning analysis framework for complex wafer bin map classification. *IEEE Transactions on Semiconductor Manufacturing*, *36*(3), 367–377.

Wang, Y., Ni, D., & Huang, Z. (2023). A momentum contrastive learning framework for low-data wafer defect classification in semiconductor manufacturing. *Applied Sciences*, *13*(5894).

Wang, J., Xu, C., et al. (2020). Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition. *IEEE Transactions on Semiconductor Manufacturing*, *33*(4), 587–596.

Weinberger, K., & Saul, L. (2004). Unsupervised learning of image manifolds by semidefinite programming. In *IEEE conf. computer vision and pattern recognition* (pp. 988–995).

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.

Wu, Z., Xiong, Y., Yu, S., & Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *IEEE conference on computer vision and pattern recognition*.

Yu, J., & Lu, X. (2016). Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis. *IEEE Transactions on Semiconductor Manufacturing*, *29*(1), 33–43.

Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. In *European conference on computer vision* (pp. 649–666). Springer.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE international conference on computer vision* (pp. 2223–2232).