Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Semi-supervised rotation-invariant representation learning for wafer map pattern analysis

Hyungu Kang, Seokho Kang [*]

Department of Industrial Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon 16419, Republic of Korea

ARTICLE INFO

Keywords:
Wafer map pattern analysis
Representation learning
Semi-supervised learning
Convolutional neural network

ABSTRACT

Recently, data-driven approaches have been widely employed to analyze the defect patterns in wafer maps, which are crucial for identifying the root causes of failures in the semiconductor fabrication process. Representation learning embeds wafer maps into compact vector representations of useful features, based on which various downstream tasks can be performed to efficiently analyze the patterns on a large scale. If wafer maps are annotated with their defect class labels, the learned representations of wafer maps will be more informative and discriminative in defect patterns. However, the manual labeling of all wafer maps by domain experts is difficult due to practical constraints. In this study, we present a semi-supervised representation learning method that fully utilizes the information from both unlabeled and labeled wafer maps to learn better representations of wafer maps with a lower labeling cost. Given a partially labeled dataset, rotation-invariant representations of wafer maps are learned using the following three objectives. First, each unlabeled wafer map is close to any wafer map of a certain class and far from those of other classes. Second, each pair of labeled wafer maps are close to each other if they belong to the same class and are far from each other otherwise. Third, the different rotations of each wafer map are close to each other for both the unlabeled and labeled wafer maps. The effectiveness of the proposed method is demonstrated for various downstream tasks related to wafer map pattern analysis: visualization, clustering, retrieval, and classifier training.

1. Introduction

In the semiconductor fabrication process, wafers undergo several complicated processing steps. After wafer fabrication is complete, all dies in a wafer are examined in terms of their electrical properties (Mann et al., 2004). The results indicating whether each die passes the test are recorded in a two-dimensional arrangement, called a wafer map. The wafer map shows the spatial information of defective dies in the fabricated wafer. Notably, defect patterns are associated with specific problems during the fabrication process (Hansen et al., 1997; Yuan et al., 2011). Thus, if a defect pattern occurs repeatedly in multiple wafer maps, engineers can determine the root cause and take corrective measures immediately.

Recently, data-driven approaches have been widely used for efficient analysis of wafer map defect patterns on a large scale. They commonly require the transformation of a raw wafer map into a vector representation to perform downstream tasks, such as visualization, clustering, retrieval, and classifier training. To ensure a better performance in such downstream tasks, the vector representation of a wafer map should be informative with respect to the corresponding defect pattern.

A classical approach for obtaining vector representations is manual feature extraction (Wu et al., 2014; Saqlain et al., 2019; Piao et al., 2018). Herein, the feature extraction rules are manually designed by domain experts, and this task requires a high level of domain knowledge. Using these rules, a wafer map is transformed into a vector of handcrafted features. The commonly used features include density, geometry, and radon-based features. The major drawback of this approach is that the representation quality is highly dependent on the feature extraction rules. The representation may lose some important information from the original wafer map.

With the advent of deep learning, a representation learning approach has been used to automatically extract useful features in a data-driven fashion without the need of manual feature extraction (Bengio et al., 2013; Le-Khac et al., 2020). Because a raw wafer map is represented as a two-dimensional array, a convolutional neural network (CNN) can be used to learn representations of wafer maps. After a CNN is trained using a set of previously collected wafer maps, it can be used to embed a new wafer map into a vector representation. This approach has proven effective in obtaining more compact and informative representations compared to the manual feature extraction approach.

Several studies have focused on unsupervised representation learning, because unlabeled wafer maps are considerably easier and cheaper

[*] Corresponding author.
E-mail addresses: viamonoh@skku.edu (H. Kang), s.kang@skku.edu (S. Kang).

https://doi.org/10.1016/j.engappai.2023.105864
Received 23 May 2022; Received in revised form 17 December 2022; Accepted 14 January 2023
Available online 20 January 2023

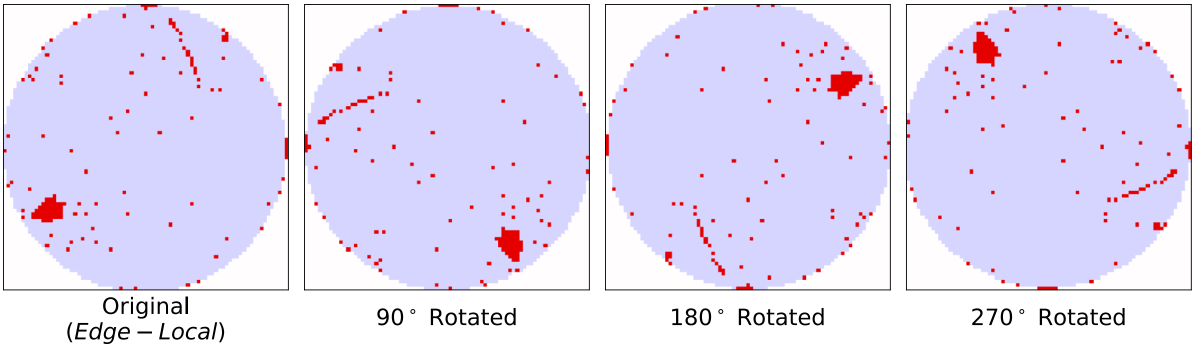


Fig. 1. Examples of rotated wafer maps.

whereas manual labeling of wafer maps is labor-intensive and time-consuming (Le-Khac et al., 2020; Zhai et al., 2019). If wafer maps are annotated with their defect class labels, a supervised representation learning approach can be employed to learn more informative and discriminative representations of wafer maps with respect to defect patterns. However, large-scale annotation of every wafer map is difficult in practice due to the limited budget and time. To address the issue, a semi-supervised representation learning approach that fully utilizes the information from both unlabeled and labeled wafer maps can be a compromising solution.

In this study, we propose a semi-supervised rotation-invariant learning method. It uses both unlabeled and labeled wafer maps to learn representations that are discriminative in defect patterns in a cost-efficient manner. In addition, it uses the rotational invariance property of a wafer map with respect to its defect class. Generally, rotation is a label-preserving transformation operation for wafer maps. Fig. 1 presents an example of a wafer map and its rotations. The original wafer map belongs to the "*Edge-Local*" class. After rotating the wafer map, it can still be regarded as belonging to the same class.

Given a partially labeled training dataset of wafer maps, the proposed method builds a CNN that learns rotation-invariant representations using an objective function, which involves the following three loss terms: (1) each unlabeled wafer map and its random rotations are close to any wafer map of a certain class and far from those of the other classes; (2) each pair of labeled wafer maps and their random rotations are close to each other if they belong to the same class and are far from each other otherwise; (3) the different rotations of each wafer map are close to each other for both unlabeled and labeled wafer maps.

The proposed method allows for better representation learning of wafer maps from a partially labeled dataset with fewer labeled wafer maps, thereby lowering the labeling cost required to achieve the desired performance for downstream tasks. In our analysis, we investigated the effectiveness of the learned representation for several downstream tasks related to wafer map pattern analysis: visualization, clustering, retrieval, and classifier training.

The remainder of this paper is organized as follows. We review related work in Section 2. The proposed method is explained in Section 3. We describe the experimental settings in Section 4, and results for various downstream tasks are presented in Section 5. Finally, the conclusions are presented in Section 6.

2. Related work

2.1. Wafer map pattern analysis

In semiconductor manufacturing, wafer map pattern analysis is crucial for examining the root causes of process failures. In that regard, data-driven approaches have been actively studied in the literature for efficient analysis and can be categorized into two main approaches: clustering and classification.

Clustering is an unsupervised learning task used to group similar instances in unlabeled data. The clustering approach is useful for wafer map pattern analysis to define defect class labels from a pool of unlabeled wafer maps. Understanding a group of similar wafer maps provides clues for identifying major defect patterns. Shim et al. (2021) extracted handcrafted features from a wafer map to obtain its vector representation and performed k-means clustering on the representation to derive clusters of wafer maps. Kim and Kang (2021) used a convolutional autoencoder (CAE) to learn the representation of a wafer map and performed dynamic clustering based on a Dirichlet process Gaussian mixture model (DPGMM). Tulala et al. (2018) performed clustering on representations learned using a variational autoencoder (VAE). Further, Hwang and Kim (2020) augmented the VAE with DPGMM for one-step clustering of wafer maps.

Classification is a supervised learning task for classifying an instance into one of the pre-defined classes. The classification model is trained using a pool of labeled instances. For wafer map pattern analysis, the classification approach benefits from the use of defect class labels annotated by domain experts, with a particular intent. The classification model can be used for the automatic classification of wafer maps in real time at a low cost. Notably, previous studies (Wu et al., 2014; Saqlain et al., 2019; Piao et al., 2018) have used manual feature extraction to transform a wafer map into a vector representation, on which various off-the-shelf classifiers, such as support vector machine and decision tree, have been built. Recently, the use of CNN has led to considerable performance improvement. A CNN can be built by learning from data in an end-to-end manner, without manual feature extraction. Nakazawa and Kulkarni (2018) adopted a CNN to classify wafer map patterns for the first time without feature engineering. Kang (2020) trained a CNN using rotation-augmented data for rotation-invariant wafer map pattern classification. Kang and Kang (2021) constructed a stacking ensemble to integrate a CNN with manual feature extraction.

For the classification approach, numerous labeled wafer maps are required to achieve high classification accuracy. However, because numerous wafer maps are produced in the semiconductor manufacturing process, domain experts often deem the task of annotating defect class labels time-consuming and labor intensive. To address this issue, recent studies have presented semi-supervised learning methods to build a classification model by learning from partially labeled data. In this regard, Kong and Ni (2018) presented a ladder network with an encoder–decoder architecture to leverage the information on unlabeled wafer maps. Yang and Yu (2020) pseudo-labeled the unlabeled wafer maps whose class labels could be predicted with high confidence. Kong and Ni (2020) adapted a semi-supervised VAE for wafer map pattern classification.

Most studies on wafer map pattern analysis have presented methods specialized for clustering and classification tasks. The main aim of this study is to learn general-purpose representations of wafer maps so that they can be used for various downstream tasks.

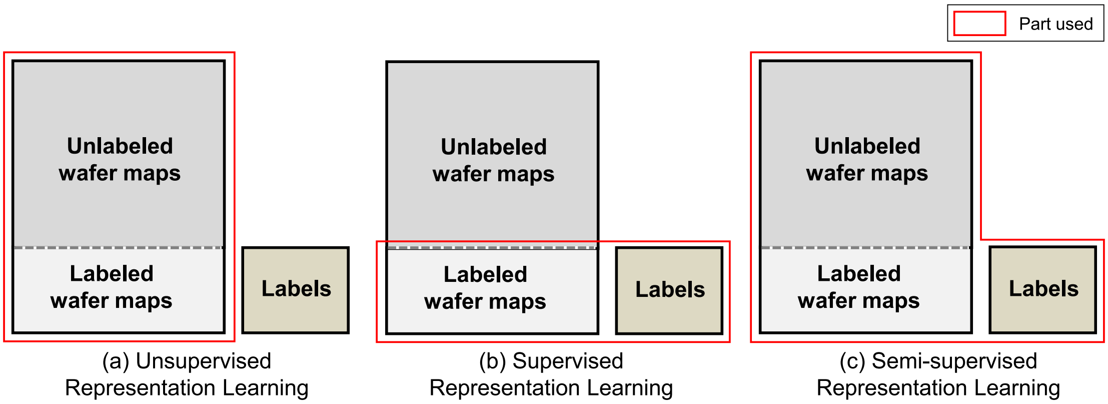

Part used
Unlabeled wafer maps
Labeled wafer maps
Labels
(a) Unsupervised Representation Learning
(b) Supervised Representation Learning
(c) Semi-supervised Representation Learning

Fig. 2. Schematic comparison of representation learning approaches.

2.2. Representation learning

Representation learning refers to the learning of parametric mapping that embeds a raw input into a vector representation (Le-Khac et al., 2020). Unlike manual feature extraction, representation learning aims to extract features automatically in a data-driven manner. Representation learning is particularly useful when the type of raw input is complex and multi-dimensional, such as images, text, and graphs. This makes it easier to extract useful information from the data to perform various downstream tasks. Representation learning can be categorized into unsupervised, supervised, and semi-supervised learning approaches according to the use of label information.

Given a set of wafer maps, among which only a small subset is labeled with defect classes, unsupervised representation learning uses all wafer maps, but discards label information, as shown in Fig. 2(a). Supervised representation learning uses only labeled wafer maps, as presented in Fig. 2(b). Semi-supervised representation learning uses the entire wafer maps and label information, as shown in Fig. 2(c).

Unsupervised representation learning refers to the learning of representations without using label information. This approach builds a model that learns from data to predict a pseudo-label through self-supervision. So far, various methods specialized for image data have been presented, including reconstruction of the original input using an autoencoder (AE) (Vincent et al., 2010), contrastive learning (Chen et al., 2020; He et al., 2020), prediction of the rotation angle (Komodakis and Gidaris, 2018), prediction of a relative patch location (Doersch et al., 2015), and solving of a jigsaw puzzle (Noroozi and Favaro, 2016).

Supervised representation learning provides more discriminative representations by leveraging the label information. Most studies have modified unsupervised representation learning methods to additionally predict the label for an input. Du et al. (2019) augmented an AE using an auxiliary softmax layer to predict class labels. The learning objective was a combination of the reconstruction and classification losses. Khosla et al. (2020) proposed a supervised contrastive learning method.

Semi-supervised representation learning effectively utilizes both unlabeled and labeled instances from a partially labeled dataset to learn the representations. In many real-world situations, there are abundant unlabeled instances; however, a few labeled instances are available for use owing to the limited budget for data labeling (Luo et al., 2017). Most existing methods obtained pseudo-labels of unlabeled instances by learning from labeled instances and then performed representation learning using the entire training dataset in a supervised manner. Revanur et al. (2021) pseudo-labeled unlabeled instances based on the visual similarity between unlabeled and labeled instances. Chen et al. (2019) predicted the pseudo-label of each unlabeled instance by using an ensemble of classifiers trained on the labeled instances. Yu et al. (2018) adopted the label graph propagation to predict the labels for unlabeled instances. Despite the effectiveness, these methods have a high risk of generating noisy labels that negatively affect the quality of learned representations if labeled instances are extremely rare in the training dataset. To avoid the need for pseudo-labeling, Hoffer and Ailon (2017) proposed a method to learn distance relations within labeled instances while imposing minimum entropy constraints on unlabeled instances.

In this study, we propose a semi-supervised rotation-invariant representation learning method to better learn informative and class-discriminative representations of wafer maps. To fully utilize the information in a partially labeled training dataset, we adapt Hoffer and Ailon (2017)'s method to semi-supervised representation learning of wafer maps. To leverage the rotational invariance property of wafer maps with respect to defect classes, we impose a rotational invariance constraint based on the notion of consistency regularization (Yang et al., 2021) for rotation-invariant representation learning of wafer maps.

3. Proposed method

3.1. Overview

The goal of representation learning for wafer maps is to build a representation model f that maps a wafer map $\mathbf{X} \in \{0,1\}^{p\times q}$ to a d-dimensional vector $\mathbf{z} \in \mathbb{R}^d$ as:

$$\mathbf{z} = f(\mathbf{X}). \tag{1}$$

Once the representation model f is trained, it can be used to transform a new wafer map into a vector representation. We can use the representation to perform various downstream tasks for wafer map pattern analysis.

In this study, we suppose that a set of wafer maps is given as the training dataset, among which only a subset is labeled with their defect classes. As stated, the proposed method follows a semi-supervised learning approach to learn the general-purpose representations of wafer maps by fully utilizing the partially labeled training dataset. We parameterize the representation model f as a CNN, for which any CNN architecture can be used, such as VGGNet and ResNet. The model f is trained using a partially labeled training dataset $\mathcal{D} = \mathcal{D}^{\mathrm{U}} \cup \mathcal{D}^{\mathrm{L}}$ containing the unlabeled part $\mathcal{D}^{\mathrm{U}} = \{\mathbf{X}_i\}_{i=1}^{N}$ and labeled part $\mathcal{D}^{\mathrm{L}} = \{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=N+1}^{N+M}$, where $\mathbf{y}_i \in \{0,1\}^C$ is a C-dimensional one-hot vector indicating the class label (*i.e.*, $\mathbf{y}_{ic} = 1$ if $\mathbf{X}_i$ belongs to the cth defect class).

The learning objective comprises three loss terms: unsupervised, supervised, and rotational invariance losses, as illustrated in Fig. 3. For unsupervised and supervised losses, we adopt the method presented in the study of Hoffer and Ailon (2017) to learn distance relations to make the representations more discriminative in defect classes. The unsupervised loss for unlabeled wafer maps is used to learn the

Algorithm 1 Training procedure of the proposed method

```
input: training dataset $\mathcal{D} = \mathcal{D}^{\mathrm{U}} \cup \mathcal{D}^{\mathrm{L}}$ (unlabeled part $\mathcal{D}^{\mathrm{U}}$ and labeled part $\mathcal{D}^{\mathrm{L}}$)
output: CNN $f$
procedure TRAIN($\mathcal{D}$)
    $f \leftarrow$ initialize a CNN
    while not termination condition do
        for each minibatch $\mathcal{S} = \mathcal{S}^{\mathrm{U}} \cup \mathcal{S}^{\mathrm{L}}$ ($\mathcal{S}^{\mathrm{U}} \subset \mathcal{D}^{\mathrm{U}}$ and $\mathcal{S}^{\mathrm{L}} \subset \mathcal{D}^{\mathrm{L}}$) do
            $\mathbf{R}_c \leftarrow$ a wafer map randomly sampled from $\{\mathbf{X}_i | (\mathbf{X}_i, \mathbf{y}_i) \in \mathcal{D}^{L}, \mathbf{y}_{ic} = 1\}$, $c = 1, \ldots, C$
            $\tilde{\mathcal{R}} \leftarrow \{\tilde{\mathbf{R}}_1, \ldots, \tilde{\mathbf{R}}_C\}$, where $\tilde{\mathbf{R}}_c$ is a rotation variant of $\mathbf{R}_c$
            ▷ objective function for unsupervised learning
            $\mathcal{J}_{\mathrm{u}} \leftarrow \frac{1}{|\mathcal{S}^{\mathrm{U}}|} \sum_{\mathbf{X}_i \in \mathcal{S}^{\mathrm{U}}} \mathcal{L}_{\mathrm{u}}(\tilde{\mathbf{X}}_i; \tilde{\mathcal{R}})$, where $\tilde{\mathbf{X}}_i$ is a rotation variant of $\mathbf{X}_i$
            ▷ objective function for supervised learning
            $\mathcal{J}_{\mathrm{s}} \leftarrow \frac{1}{|\mathcal{S}^{\mathrm{L}}|} \sum_{(\mathbf{X}_i, \mathbf{y}_i) \in \mathcal{S}^{\mathrm{L}}} \mathcal{L}_{\mathrm{s}}(\tilde{\mathbf{X}}_i, \mathbf{y}_i; \tilde{\mathcal{R}})$, where $\tilde{\mathbf{X}}_i$ is a rotation variant of $\mathbf{X}_i$
            ▷ objective function for rotation-invariant learning
            $\mathcal{J}_{\mathrm{rot}} \leftarrow \frac{1}{|\mathcal{S}^{\mathrm{U}}|+|\mathcal{S}^{\mathrm{L}}|} \left[ \sum_{\mathbf{X}_i \in \mathcal{S}^{\mathrm{U}}} \mathcal{L}_{\mathrm{rot}}(\mathbf{X}_i) + \sum_{(\mathbf{X}_i, \mathbf{y}_i) \in \mathcal{S}^{\mathrm{L}}} \mathcal{L}_{\mathrm{rot}}(\mathbf{X}_i) \right]$
            ▷ total objective function for semi-supervised rotation-invariant learning
            $\mathcal{J} \leftarrow \mathcal{J}_{\mathrm{u}} + \mathcal{J}_{\mathrm{s}} + \lambda \cdot \mathcal{J}_{\mathrm{rot}}$
            ▷ training the representation model
            $f \leftarrow$ update the parameters of $f$ by gradient descent of $\mathcal{J}$
        end for
    end while
    return $f$
end procedure
```

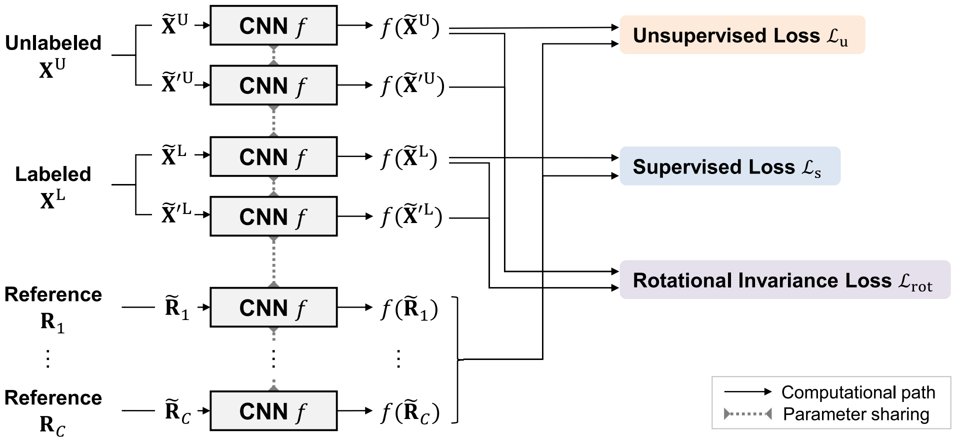

Unlabeled $\mathbf{X}^{\mathrm{U}}$
$\tilde{\mathbf{X}}^{\mathrm{U}}$
CNN f
$f(\tilde{\mathbf{X}}^{\mathrm{U}})$
$\tilde{\mathbf{X}}'^{\mathrm{U}}$
CNN f
$f(\tilde{\mathbf{X}}'^{\mathrm{U}})$
Labeled $\mathbf{X}^{\mathrm{L}}$
$\tilde{\mathbf{X}}^{\mathrm{L}}$
CNN f
$f(\tilde{\mathbf{X}}^{\mathrm{L}})$
$\tilde{\mathbf{X}}'^{\mathrm{L}}$
CNN f
$f(\tilde{\mathbf{X}}'^{\mathrm{L}})$
Reference $\mathbf{R}_1$
$\tilde{\mathbf{R}}_1$
CNN f
$f(\tilde{\mathbf{R}}_1)$
Reference $\mathbf{R}_C$
$\tilde{\mathbf{R}}_C$
CNN f
$f(\tilde{\mathbf{R}}_C)$
Unsupervised Loss $\mathcal{L}_{\mathrm{u}}$
Supervised Loss $\mathcal{L}_{\mathrm{s}}$
Rotational Invariance Loss $\mathcal{L}_{\mathrm{rot}}$
Computational path
Parameter sharing

Fig. 3. Loss functions for semi-supervised rotation-invariant learning.

representations to be close to a specific class and far from other classes. The supervised loss for labeled wafer maps is used to learn the representations of wafer maps from the same class to be close to each other and those from different classes to be far from each other. The rotational invariance loss is used to make the model utilize the rotational invariance property of wafer maps with respect to defect classes, which makes the representations invariant to rotations of both unlabeled and labeled wafer maps.

Algorithm 1 presents the pseudocode of the training procedure of the proposed method. The detail of each loss term is described in the following subsection.

3.2. Learning objective

At each training iteration, we use a minibatch $\mathcal{S} = \mathcal{S}^{\mathrm{U}} \cup \mathcal{S}^{\mathrm{L}}$, where $\mathcal{S}^{\mathrm{U}}$ and $\mathcal{S}^{\mathrm{L}}$ denote the unlabeled and labeled subsets sampled from $\mathcal{D}^{\mathrm{U}}$ and $\mathcal{D}^{\mathrm{L}}$, respectively. In addition, we randomly sample one wafer map per class from $\mathcal{D}^{\mathrm{L}}$ to constitute a reference set $\mathcal{R} = \{\mathbf{R}_1, \ldots, \mathbf{R}_C\}$, where $\mathbf{R}_c$ belongs to the cth class.

For each wafer map $\mathbf{X}$ in the minibatch $\mathcal{S}$, we define a C-dimensional relation vector $P(\mathbf{X}; \mathcal{R})$, which represents its distance relation to each element $\mathbf{R}_c$ in the reference set $\mathcal{R}$, whose cth element $P(\mathbf{X}; \mathcal{R})_c$ is computed as follows:

$$P(\mathbf{X}; \mathcal{R})_c = \frac{e^{-\|f(\mathbf{X})-f(\mathbf{R}_c)\|^2}}{\sum_{j=1}^{C} e^{-\|f(\mathbf{X})-f(\mathbf{R}_j)\|^2}}. \tag{2}$$

This can be interpreted as the probability estimate of $\mathbf{X}$ being classified into each class (Hoffer and Ailon, 2017).

For the model f to learn from various rotation variants of wafer maps, we randomly rotate each wafer map by a rotation angle sampled from the uniform distribution $\mathcal{U}(0, 360°)$ and randomly flip it with a probability of 0.5. The rotation variants of wafer maps $\mathbf{X}$ in the minibatch $\mathcal{S}$ and $\mathbf{R}_c$ in the reference set $\mathcal{R}$ are respectively denoted as $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{R}}_c$, and $\tilde{\mathcal{R}}$ is defined as $\{\tilde{\mathbf{R}}_1, \ldots, \tilde{\mathbf{R}}_C\}$. Because rotation is a label-preserving transformation, each rotation variant has the same class label as the original (Kang, 2020).

The unsupervised loss term $\mathcal{L}_{\mathrm{u}}$ is used for unsupervised representation learning of the representation model f from unlabeled wafer maps. For an unlabeled wafer map $\mathbf{X}$, the loss term is defined as the entropy of the relation vector $P(\mathbf{X};\mathcal{R})$:

$$\begin{aligned}\mathcal{L}_{\mathrm{u}}(\mathbf{X};\mathcal{R}) &= \mathcal{H}(P(\mathbf{X};\mathcal{R})) \\ &= -\sum_{c=1}^{C} P(\mathbf{X};\mathcal{R})_c \cdot \log P(\mathbf{X};\mathcal{R})_c.\end{aligned} \tag{3}$$

With $\mathcal{L}_{\mathrm{u}}$, the model f learns to map $\mathbf{X}$ to be close to any one wafer map in $\mathcal{R}$ and to be far from the others in $\mathcal{R}$.

The supervised loss term $\mathcal{L}_{\mathrm{s}}$ is used for supervised representation learning of the representation model f from labeled wafer maps. For a labeled wafer map $(\mathbf{X},\mathbf{y})$, the loss term is defined as the cross-entropy between the one-hot label vector $\mathbf{y}$ and the relation vector $P(\mathbf{X};\mathcal{R})$:

$$\begin{aligned}\mathcal{L}_{\mathrm{s}}(\mathbf{X},\mathbf{y};\mathcal{R}) &= \mathcal{H}(\mathbf{y},P(\mathbf{X};\mathcal{R})) \\ &= -\sum_{c=1}^{C} y_c \cdot \log P(\mathbf{X};\mathcal{R})_c.\end{aligned} \tag{4}$$

With $\mathcal{L}_{\mathrm{s}}$, the model f learns to map $\mathbf{X}$ to be close to the wafer map in $\mathcal{R}$ that belongs to the same class and to be far from the others in $\mathcal{R}$.

The rotational invariance term $\mathcal{L}_{\mathrm{rot}}$ is used for rotation-invariant representation learning of the representation model f from both unlabeled and labeled wafer maps. It encourages the model f to produce the same representation when the input wafer map is rotated. For a wafer map $\mathbf{X}$, the loss term is used to explicitly impose a constraint on the representation model f that the rotation variants of a wafer map should have similar representations (*i.e.*, $f(\mathbf{X}) \simeq f(\tilde{\mathbf{X}})$). This term is defined as the Euclidean distance between the representations of the two rotation variants of $\mathbf{X}$:

$$\mathcal{L}_{\mathrm{rot}}(\mathbf{X}) = \|f(\tilde{\mathbf{X}}) - f(\tilde{\mathbf{X}}')\|^2, \tag{5}$$

where $\mathbf{X}'$ is a copy of $\mathbf{X}$ such that $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{X}}'$ are two different rotation variants of $\mathbf{X}$. The addition of $\mathcal{L}_{\mathrm{rot}}$ allows the model f to learn the rotation-invariant representations of the wafer maps.

The total objective function $\mathcal{J}$ for semi-supervised rotation-invariant representation learning is a combination of the three loss terms $\mathcal{L}_{\mathrm{u}}$, $\mathcal{L}_{\mathrm{s}}$, and $\mathcal{L}_{\mathrm{rot}}$ computed on the minibatch $\mathcal{S}$. For each loss term, the rotation variants of wafer maps are used as inputs so that the representations are learned from various rotation variants of wafer maps. The function $\mathcal{J}$ is described as:

$$\begin{aligned}\mathcal{J}(f) =&\mathcal{J}_{\mathrm{u}} + \mathcal{J}_{\mathrm{s}} + \lambda \cdot \mathcal{J}_{\mathrm{rot}} \\ =&\frac{1}{|\mathcal{S}^{\mathrm{U}}|}\sum_{\mathbf{X}_i \in \mathcal{S}^{\mathrm{U}}} \mathcal{L}_{\mathrm{u}}(\tilde{\mathbf{X}}_i;\tilde{\mathcal{R}}) + \frac{1}{|\mathcal{S}^{\mathrm{L}}|}\sum_{(\mathbf{X}_i,\mathbf{y}_i) \in \mathcal{S}^{\mathrm{L}}} \mathcal{L}_{\mathrm{s}}(\tilde{\mathbf{X}}_i,\mathbf{y}_i;\tilde{\mathcal{R}}) \\ &+ \frac{\lambda}{|\mathcal{S}^{\mathrm{U}}| + |\mathcal{S}^{\mathrm{L}}|}\left[\sum_{\mathbf{X}_i \in \mathcal{S}^{\mathrm{U}}} \mathcal{L}_{\mathrm{rot}}(\mathbf{X}_i) + \sum_{(\mathbf{X}_i,\mathbf{y}_i) \in \mathcal{S}^{\mathrm{L}}} \mathcal{L}_{\mathrm{rot}}(\mathbf{X}_i)\right],\end{aligned} \tag{6}$$

where $\mathcal{J}_{\mathrm{u}}$, $\mathcal{J}_{\mathrm{s}}$, and $\mathcal{J}_{\mathrm{rot}}$ are the objective functions for supervised learning, unsupervised learning, and rotation-invariant learning, respectively, and λ is a hyperparameter that controls the strength of the rotational invariance constraint. During the training, the parameters of the model f are updated toward minimizing the function $\mathcal{J}$.

4. Experimental settings

4.1. Data description

The WM-811k dataset (Wu et al., 2014) was used to evaluate the performance of the proposed method. The dataset consists of 811,457 wafer maps collected from a semiconductor manufacturer. Among them, 172,950 wafer maps were labeled with one of the following nine defect classes by domain experts: *None*, *Center*, *Donut*, *Edge-Local*, *Edge-Ring*, *Local*, *Random*, *Scratch*, and *Near-Full*. Fig. 4 presents examples of wafer maps according to the defect class.

For the experiments, we preprocessed the dataset as follows. We used only the labeled part of the dataset. Four abnormal wafer maps with fewer than 100 dies were excluded. The sizes of the wafer maps ranged between 6 × 21 and 300 × 202. Each wafer map was resized to 64 × 64. Notably, the original dataset had a highly imbalanced class distribution, with the *None* class accounting for greater than 85% of the dataset. Thus, we reduced the size of the *None* class to 1/10 using random undersampling. The preprocessed dataset had a total of 40,261 labeled wafer maps. Table 1 lists the class distributions of the dataset.

Table 1
Class distribution of the dataset used in the experiments.

Index (c)	Defect class	No. wafer maps	Ratio (%)
1	*None*	14,742	36.62
2	*Center*	4,294	10.67
3	*Donut*	555	1.38
4	*Edge-Local*	5,189	12.89
5	*Edge-Ring*	9,680	24.04
6	*Local*	3,593	8.92
7	*Random*	866	2.15
8	*Scratch*	1,193	2.96
9	*Near-Full*	149	0.37
	Total	40,261	100.00

For the experiments, 90% of the dataset was used to form the training dataset $\mathcal{D}$ for representation learning and the remaining 10% of the dataset was used as the test set for performance evaluation of downstream tasks. For the training dataset $\mathcal{D}$, we varied the proportions of labeled wafer maps as 1%, 2%, 5%, 10%, 20%, 50%, and 100%. For example, if the labeled ratio was set to 5% (*i.e.*, $|\mathcal{D}^L|/|\mathcal{D}| = 0.05$), we randomly sampled 5% of the wafer maps from $\mathcal{D}$ and used them as the labeled part $\mathcal{D}^L$ whereas the remaining 95% was used as the unlabeled part $\mathcal{D}^U$ by eliminating the labels.

4.2. Compared methods

The proposed method, semi-supervised rotation-invariant representation learning (**SSRL+Rot**), was compared with eight baseline methods. Each baseline method was characterized by learning representations, the use of label information, the use of unlabeled wafer maps, and consideration of rotational invariance. Table 2 summarizes the methods used in these experiments. Additional details of each process are described below:

- **Manual feature extraction (MFE)** obtains a vector representation of handcrafted features for a wafer map without any learning procedure (Wu et al., 2014). For the handcrafted features, we use 13 density, 40 geometry, and 6 radon features. Accordingly, the representation is 59-dimensional. Some of these features exhibit the rotational invariance property. From a computational perspective, MFE is the most efficient in terms of both training and inference because it does not use learned representations but simply computes manual features only. In the experiments, we standardized each feature to have a mean of zero and variance of one on the training dataset.
- **Unsupervised representation learning with CAE** (**URL$_{\mathbf{CAE}}$**) learns the representations of wafer maps through a CAE (Kim and Kang, 2021). We employ a CAE, whose encoder is set to be the same as that of the proposed method, and decoder is set to the inverse of the encoder. The model is trained without using label information in the training dataset. Once trained, the encoder is used as the representation model.
- **Unsupervised rotation-invariant representation learning with CAE** (**URL+Rot$_{\mathbf{CAE}}$**) is an extension of **URL$_{\mathbf{CAE}}$** for learning rotation-invariant representations. During training of the model, rotation-based data augmentation is applied to the training dataset.
- **Unsupervised rotation-invariant representation learning with SimCLR** (**URL+Rot$_{\mathbf{SimCLR}}$**) obtains the representations of wafer maps through contrastive learning based on the simple framework for contrastive learning (SimCLR) (Chen et al., 2020).

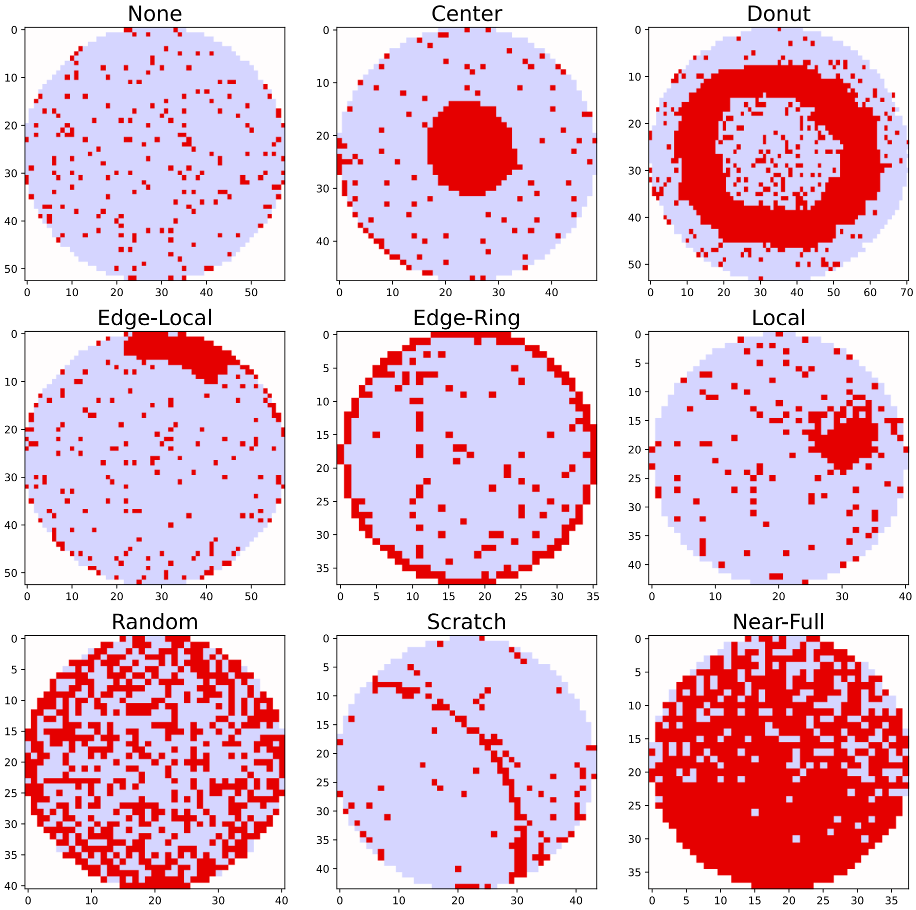


Fig. 4. Examples of wafer maps in the WM-811K dataset.

Table 2
Summary of compared methods.

Method	Learned representation	Use label information	Use unlabeled wafer maps	Rotational invariance
MFE	X	X	X	△
URL$_{\text{CAE}}$	O	X	O	X
URL+Rot$_{\text{CAE}}$	O	X	O	O
URL+Rot$_{\text{SimCLR}}$	O	X	O	O
URL+Rot$_{\text{MoCo}}$	O	X	O	O
SRL	O	O	X	X
SRL+Rot	O	O	X	O
SSRL	O	O	O	X
SSRL+Rot (Proposed)	O	O	O	O

The encoder of the model architecture is set to be the same as that of the proposed method. A projection head is attached to the top of the encoder. The model is trained to maximize the agreement between different augmented views of the same instance in the projected representation. Similar to the proposed method, we use the rotation variants of a wafer map to create the augmented views. Once trained, the encoder is used as the representation model.

- **Unsupervised rotation-invariant representation learning with MoCo** (**URL+Rot$_{\text{MoCo}}$**) learns the representations of wafer maps based on the momentum contrast method (MoCo) (He et al., 2020). It trains a representation model by maximizing the agreement of representations between random augmented views of an instance and minimizing that between the instance and all other instances in a dictionary, during which the dictionary is dynamically updated by adding the representations of the current minibatch and removing those of the oldest minibatch.
- **Supervised representation learning (SRL)** is an ablation of the proposed method that uses the supervised loss term only as the learning objective without using rotation variants of the wafer maps. The representation model is trained using the labeled part of the training dataset.
- **Supervised rotation-invariant representation learning (SRL+Rot)** is an extension of **SRL** used to learn rotation-invariant representations. It utilizes the rotation variants of wafer maps in the supervised loss term and appends the rotational invariance term to the learning objective.
- **Semi-supervised representation learning (SSRL)** is an ablation of the proposed method that excludes the rotational invariance term from the learning objective and does not use the rotation variants of wafer maps in the supervised and unsupervised loss terms.

The computational efficiency of representational learning methods in the training phase depends on the size of training dataset and model

Table 3
Architecture of the representation model used in this study.

Layer	Activation	Shape
Input (Wafer map)	–	(64,64,3)
Convolutional Layers - Convolution with 64 Filters (3 × 3) - Convolution with 64 Filters (3 × 3) - Max-Pooling (2 × 2)	ReLU	(32,32,64)
Convolutional Layers - Convolution with 128 Filters (3 × 3) - Convolution with 128 Filters (3 × 3) - Max-Pooling (2 × 2)	ReLU	(16,16,128)
Convolutional Layers - Convolution with 256 Filters (3 × 3) - Convolution with 256 Filters (3 × 3) - Convolution with 256 Filters (3 × 3) - Convolution with 256 Filters (3 × 3) - Max-Pooling (2 × 2)	ReLU	(8,8,256)
Convolutional Layers - Convolution with 512 Filters (3 × 3) - Convolution with 512 Filters (3 × 3) - Convolution with 512 Filters (3 × 3) - Convolution with 512 Filters (3 × 3) - Max-Pooling (2 × 2)	ReLU	(4,4,512)
Convolutional Layers - Convolution with 512 Filters (3 × 3) - Convolution with 512 Filters (3 × 3) - Convolution with 512 Filters (3 × 3) - Convolution with 512 Filters (3 × 3) - Max-Pooling (2 × 2)	ReLU	(2,2,512)
Global average pooling	–	512
Fully-connected layer (512)	ReLU	512

architecture used. The time complexity for training a neural network is linear with respect to the size of the training dataset ($N + M$ for URL and SSRL methods and M for SRL methods), the number of hidden layers in the neural network, and the number of training epochs. The methods based on CAE, URL$_{\mathrm{CAE}}$ and URL+Rot$_{\mathrm{CAE}}$, require the training of an encoder–decoder model, which is computationally more expensive compared to the other methods. For URL+Rot$_{\mathrm{SimCLR}}$, an auxiliary projection head needs to be trained along with the representation model. However, URL+Rot$_{\mathrm{MoCo}}$, SRL, SRL+Rot, SSRL, and SSRL+Rot do not require the training of any auxiliary components other than the representation model.

The computational efficiency of representational learning methods in the inference phase depends on the architecture of the representation model. Because all methods use the same architecture for their representation models, the time complexity is identical.

4.3. Training settings

For the representation model f, we used modified VGG19; the architecture of this model is described in Table 3. From the original VGG19 (Simonyan and Zisserman, 2014), we excluded all the fully-connected layers and replaced the flatten operation with global average pooling. We then appended a fully-connected layer with 512 dimensions to the output layer. Thus, the dimensionality of all learned representations was 512.

For the proposed method and its ablations, the value of the hyperparameter λ in the objective function $\mathcal{J}$ was set to one. For all methods, training was performed for 30 epochs using the Adam optimizer with a minibatch size of 256, learning rate of 10^{-4}, and weight decay factor of 10^{-6}.

4.4. Evaluation settings

The representation learning performance of each method was evaluated for four downstream tasks: t-stochastic neighbor embedding (t-SNE) visualization, class separation, k-nearest neighbor retrieval, and classifier training. All experiments, except for t-SNE visualization, were performed in five independent replicates with different random seeds. Detailed settings used for the downstream tasks are described in the following section.

To assess the statistical significance of the results for each downstream task, we performed a paired t-test comparing the best-performing method with the second-best method, with a significance level of 0.05.

5. Results and discussion

5.1. t-SNE visualization

We qualitatively assessed the ability of each method to learn the class-discriminative representations by visualizing the representations in two-dimensional space. We used t-SNE (Van der Maaten and Hinton, 2008) to reduce the dimensionality to two while preserving the neighbor structure between the original representations.

Fig. 5 illustrates two-dimensional visualization results for the compared methods when 5% of the training dataset was labeled. The proposed method yielded much better representations with distinct boundaries between classes, indicating that utilizing unlabeled wafer maps and rotational invariance property in representation learning was beneficial. For the baseline methods that did not use label information (MFE, URL$_{\mathrm{CAE}}$, URL+Rot$_{\mathrm{CAE}}$, URL+Rot$_{\mathrm{SimCLR}}$, and URL+Rot$_{\mathrm{MoCo}}$), we observed a high level of overlap between the representations of different classes, making it difficult to distinguish between classes. The baseline methods that used label information (SRL, SRL+Rot, SSRL) reduced the overlap and thus the representations were more discriminative with respect to classes.

5.2. Class separation

For the class separation evaluation, we quantitatively measured how distinctively the classes were separated in the representations. We used the Davies–Bouldin index (Davies and Bouldin, 1979) as the measure of class separation on the representations for quantitative assessment. The Davies–Bouldin index was originally proposed to evaluate the clustering results, evaluating the compactness of each cluster and its separation from other clusters. A lower score indicates a better overall separation of clusters. We obtained this measure for the test dataset by assuming that each class to be a cluster.

Fig. 6 compares the Davies–Bouldin index value between the baseline and proposed methods by varying the proportion of labeled wafer maps. The results quantitatively demonstrate that the proposed method could learn more class-distinctive representations of wafer maps from a partially labeled training dataset. Overall, the proposed method demonstrated lower or comparable scores compared with the baselines. In particular, we observed that the proposed method was superior to all baselines with statistical significance when the labeled ratio was below 5%. For the methods that used label information, the performance gradually improved with the labeled ratio. By contrast, methods that did not use label information performed worse, with much higher scores compared to MFE.

5.3. k-NN retrieval

In a wafer map analysis, searching previously analyzed wafer maps with similar defect patterns helps in identifying related problems. Given a query wafer map whose class label is unknown, we can retrieve k-nearest neighbor (k-NN) wafer maps in the training dataset whose representations were close to the query wafer map in terms of the Euclidean distance. Then, we can use the class labels of the retrieved wafer maps to predict the class label of the query wafer map.

For the evaluation purpose, we simulated the k-NN retrieval procedure by using each wafer map in the test dataset as a query wafer map.

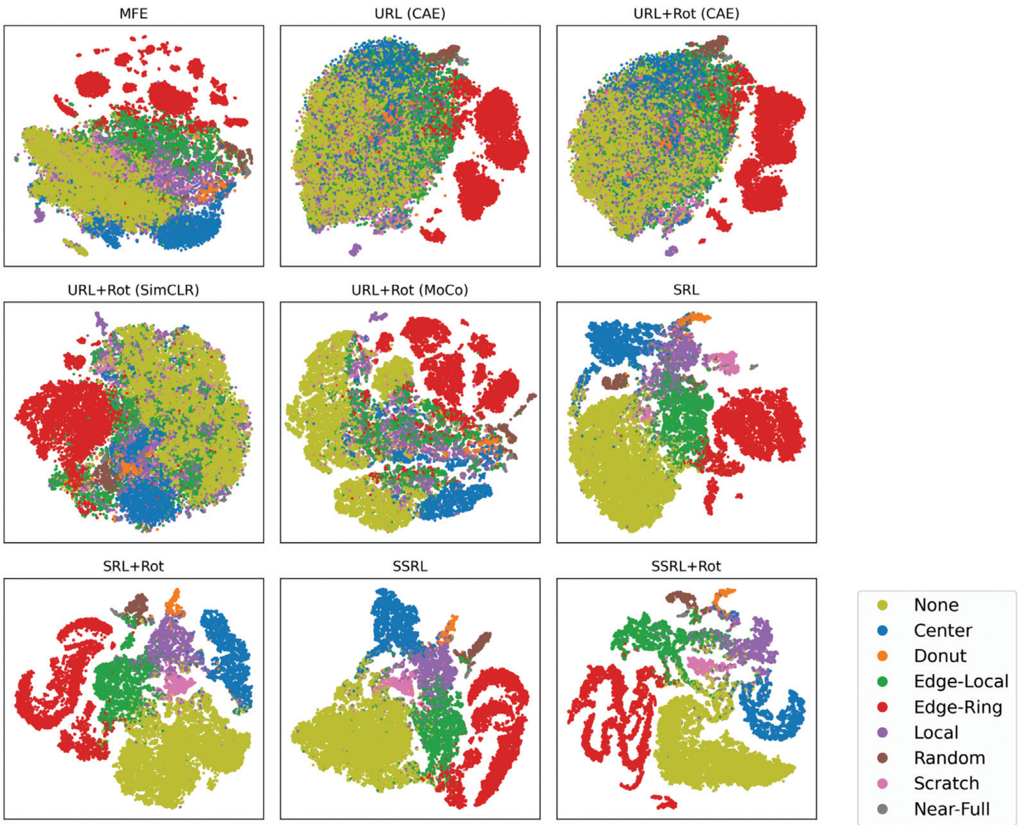


Fig. 5. Comparison of t-SNE visualization between the baseline and proposed methods (proportion of labeled wafer maps = 5%).

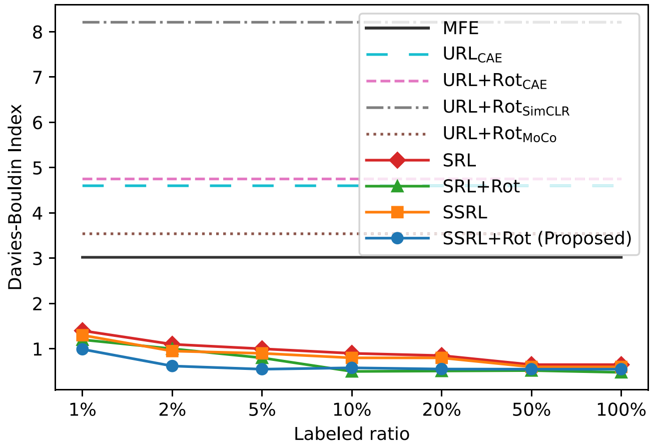


Fig. 6. Comparison of class separation (in terms of Davies–Bouldin index) between the baseline and proposed methods.

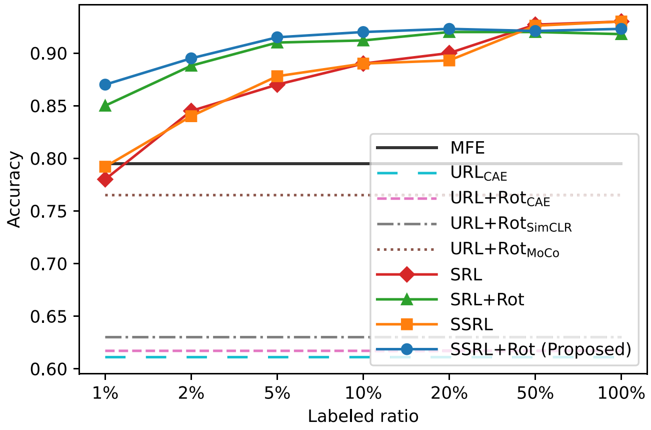


Fig. 7. Comparison of the k-NN retrieval accuracy between the baseline and proposed methods.

We assessed whether the retrieved wafer maps belong to the same class using the k-NN retrieval accuracy, which is the fraction of retrieved wafer maps that belong to the same class as the query wafer map. We computed the average k-NN retrieval accuracy on the test dataset by setting $k = 10$.

Fig. 7 depicts the k-NN retrieval accuracy of the compared methods. The proposed method significantly outperformed the baseline methods when the labeled ratio was less than 10%. Among the baselines, SRL+Rot was comparable to the proposed method when the labeled ratio was relatively high, indicating that a consideration of rotational invariance property contributed significantly to the performance improvement. However, the methods that did not use label information performed much worse than the MFE.

Fig. 8 presents an example of the k-NN retrieval results for a query wafer map obtained using SSRL and the proposed method trained with a 5% labeled dataset. The query wafer map had local defects in the upper edge region and was accordingly labeled with the *Edge-Local* class. In both methods, all retrieved wafer maps belonged to the same class as the query wafer map. All wafer maps retrieved by SSRL had local defects in the upper edge region, similar to the query wafer map. In contrast, the wafer maps retrieved by the proposed method presented local defects in different edge regions owing to the rotational invariance of the representations of wafer maps.

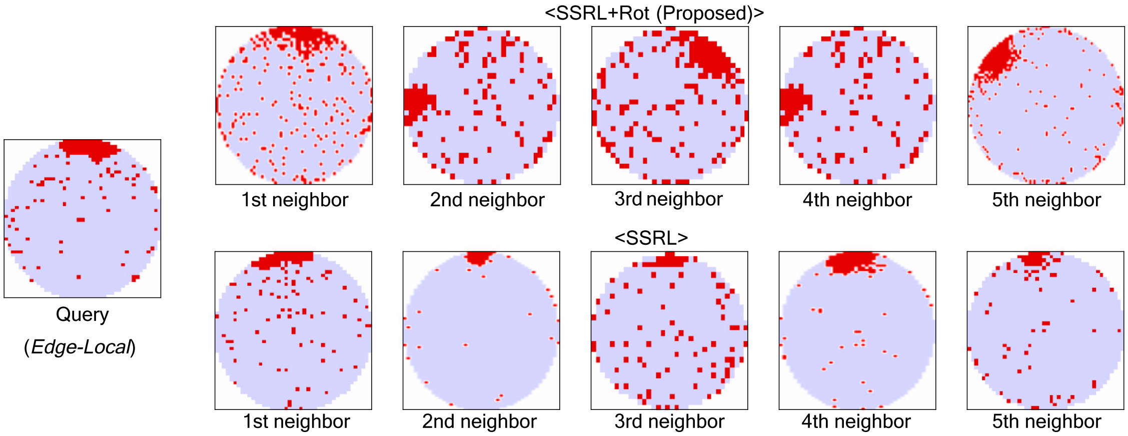
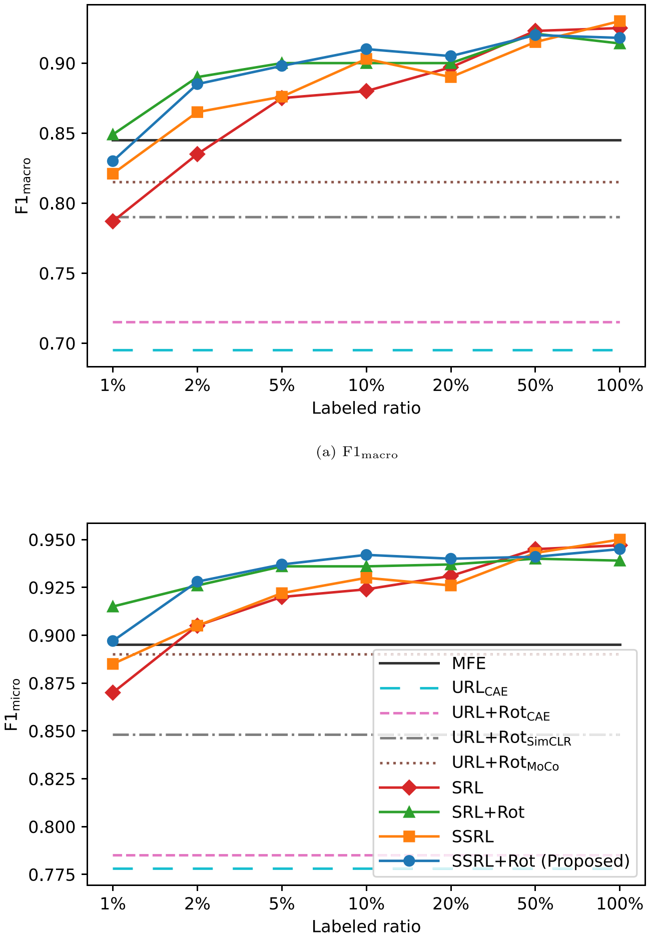


Fig. 8. Example of k-NN retrieval results yielded by the SSRL and the proposed method.



(a) $\mathrm{F1}_{\mathrm{macro}}$

(b) $\mathrm{F1}_{\mathrm{micro}}$

Fig. 9. Comparison of softmax classifier performance (in terms of $\mathrm{F1}_{\mathrm{macro}}$ and $\mathrm{F1}_{\mathrm{micro}}$) between the baseline and proposed methods.

5.4. Classifier training

Following the linear evaluation protocol, we trained a softmax classifier on the representations and evaluated its classification performance. Note that this evaluation protocol has been widely used to evaluate the representation learning performance in the literature (Chen et al., 2020; He et al., 2020). For training, the objective function was set as categorical cross-entropy. We used 90% of the training dataset for parameter update and 10% for validation. The training was terminated if the validation loss did not increase for 20 consecutive epochs. The other training settings were set as described in Section 4.3. The classification performance of the softmax classifier was evaluated in terms of the macro-average F1 score ($\mathrm{F1}_{\mathrm{macro}}$) and micro-average F1 score ($\mathrm{F1}_{\mathrm{micro}}$) on the test dataset. Here, $\mathrm{F1}_{\mathrm{macro}}$ computes an unweighted average of F1 score per class, for which the majority classes can dominate under a class imbalance. $\mathrm{F1}_{\mathrm{micro}}$ computes the F1 score using the total number of true positives, false negatives, and false positives across all classes and thus is equivalent to the simple classification accuracy.

Fig. 9 presents the classification performance of the compared methods in terms of $\mathrm{F1}_{\mathrm{macro}}$ and $\mathrm{F1}_{\mathrm{micro}}$. The overall tendencies appear to be similar to those of the k-NN retrieval evaluation results. The proposed method and SRL+Rot exhibited superior performance with statistical significance at low labeled ratios, indicating that rotation-invariant learning was beneficial when a few labeled wafer maps were available in the training dataset. The performance gradually improved with an increase in the labeled ratio. When the ratio was greater than 50%, SRL and SRL+Rot performed slightly better than the proposed method.

6. Conclusion

In this paper, we presented a semi-supervised rotation-invariant representation learning method to learn better representations of wafer maps from a partially labeled training dataset. The representation model was trained using a learning objective that involves unsupervised, supervised, and rotational invariance losses. We demonstrated that the proposed method improved performance on various downstream tasks related to wafer map pattern analysis, particularly when only a few labeled wafer maps were available in the training dataset.

The main contributions of this study can be summarized as follows. First, the proposed method fully utilized the information from both unlabeled and labeled wafer maps for representation learning. Second, a rotational invariance constraint was appended in the learning objective to learn the rotation-invariant representations of the wafer maps. We expect that the proposed method will enable efficient learning of more informative and class-discriminative representations of wafer maps at low labeling costs. This will be practically useful in the real-world situations where the labeling budget is limited.

As a direction for future research, we aim to extend the proposed method to update the representation model by interactively querying the labels of unlabeled wafer maps to further improve performance at minimal additional labeling cost.

CRediT authorship contribution statement

Hyungu Kang: Conceptualization, Methodology, Software, Writing – original draft. **Seokho Kang:** Conceptualization, Methodology, Investigation, Writing – review & editing.

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 publicly available.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) under Grant NRF-2020R1C1C1003232.

References

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1798–1828. http://dx.doi.org/10.1109/TPAMI.2013.50.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020. A simple framework for contrastive learning of visual representations. In: Proceedings of International Conference on Machine Learning. pp. 1597–1607.

Chen, K., Yao, L., Zhang, D., Wang, X., Chang, X., Nie, F., 2019. A semisupervised recurrent convolutional attention model for human activity recognition. IEEE Trans. Neural Netw. Learn. Syst. 31 (5), 1747–1756.

Davies, D.L., Bouldin, D.W., 1979. A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. 1 (2), 224–227. http://dx.doi.org/10.1109/TPAMI.1979.4766909.

Doersch, C., Gupta, A., Efros, A.A., 2015. Unsupervised visual representation learning by context prediction. In: Proceedings of IEEE International Conference on Computer Vision. pp. 1422–1430.

Du, F., Zhang, J., Ji, N., Hu, J., Zhang, C., 2019. Discriminative representation learning with supervised auto-encoder. Neural Process. Lett. 49 (2), 507–520. http://dx.doi.org/10.1007/s11063-018-9828-2.

Hansen, M.H., Nair, V.N., Friedman, D.J., 1997. Monitoring wafer map data from integrated circuit fabrication processes for spatially clustered defects. Technometrics 39 (3), 241–253. http://dx.doi.org/10.1080/00401706.1997.10485116.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738.

Hoffer, E., Ailon, N., 2017. Semi-supervised deep learning by metric embedding. In: Proceedings of International Conference on Learning Representations Workshop Track.

Hwang, J., Kim, H., 2020. Variational deep clustering of wafer map patterns. IEEE Trans. Semicond. Manuf. 33 (3), 466–475. http://dx.doi.org/10.1109/TSM.2020.3004483.

Kang, S., 2020. Rotation-invariant wafer map pattern classification with convolutional neural networks. IEEE Access 8, 170650–170658. http://dx.doi.org/10.1109/ACCESS.2020.3024603.

Kang, H., Kang, S., 2021. A stacking ensemble classifier with handcrafted and convolutional features for wafer map pattern classification. Comput. Ind. 129, 103450. http://dx.doi.org/10.1016/j.compind.2021.103450.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D., 2020. Supervised contrastive learning. In: Advances in Neural Information Processing Systems. 33, pp. 18661–18673.

Kim, D., Kang, P., 2021. Dynamic clustering for wafer map patterns using self-supervised learning on convolutional autoencoders. IEEE Trans. Semicond. Manuf. 34 (4), 444–454. http://dx.doi.org/10.1109/TSM.2021.3107720.

Komodakis, N., Gidaris, S., 2018. Unsupervised representation learning by predicting image rotations. In: Proceedings of International Conference on Learning Representations.

Kong, Y., Ni, D., 2018. Semi-supervised classification of wafer map based on ladder network. In: Proceedings of IEEE International Conference on Solid-State and Integrated Circuit Technology. http://dx.doi.org/10.1109/ICSICT.2018.8564982.

Kong, Y., Ni, D., 2020. A semi-supervised and incremental modeling framework for wafer map classification. IEEE Trans. Semicond. Manuf. 33 (1), 62–71. http://dx.doi.org/10.1109/TSM.2020.2964581.

Le-Khac, P.H., Healy, G., Smeaton, A.F., 2020. Contrastive representation learning: A framework and review. IEEE Access 8, 193907–193934.

Luo, M., Chang, X., Nie, L., Yang, Y., Hauptmann, A.G., Zheng, Q., 2017. An adaptive semisupervised feature analysis for video semantic recognition. IEEE Trans. Cybern. 48 (2), 648–660.

Van der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9 (11), 2579–2605.

Mann, W.R., Taber, F.L., Seitzer, P.W., Broz, J.J., 2004. The leading edge of production wafer probe test technology. In: Proceedings of IEEE International Test Conference. pp. 1168–1195. http://dx.doi.org/10.1109/TEST.2004.1387391.

Nakazawa, T., Kulkarni, D.V., 2018. Wafer map defect pattern classification and image retrieval using convolutional neural network. IEEE Trans. Semicond. Manuf. 31 (2), 309–314. http://dx.doi.org/10.1109/TSM.2018.2795466.

Noroozi, M., Favaro, P., 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In: Proceedings of European Conference on Computer Vision. pp. 69–84.

Piao, M., Jin, C.H., Lee, J.Y., Byun, J.-Y., 2018. Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features. IEEE Trans. Semicond. Manuf. 31 (2), 250–257. http://dx.doi.org/10.1109/TSM.2018.2806931.

Revanur, A., Kumar, V., Sharma, D., 2021. Semi-supervised visual representation learning for fashion compatibility. In: Proceedings of ACM Conference on Recommender Systems. pp. 463–472.

Saqlain, M., Jargalsaikhan, B., Lee, J.Y., 2019. A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing. IEEE Trans. Semicond. Manuf. 32 (2), 171–182. http://dx.doi.org/10.1109/TSM.2019.2904306.

Shim, J., Kang, S., Cho, S., 2021. Active cluster annotation for wafer map pattern classification in semiconductor manufacturing. Expert Syst. Appl. 183, 115429. http://dx.doi.org/10.1109/ACCESS.2020.3031549.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. http://dx.doi.org/10.48550/arXiv.1409.1556, arXiv preprint arXiv:1409.1556.

Tulala, P., Mahyar, H., Ghalebi, E., Grosu, R., 2018. Unsupervised wafermap patterns clustering via variational autoencoders. In: Proceedings of International Joint Conference on Neural Networks.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., Bottou, L., 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. 11 (12), 3371–3408.

Wu, M.-J., Jang, J.-S.R., Chen, J.-L., 2014. Wafer map failure pattern recognition and similarity ranking for large-scale data sets. IEEE Trans. Semicond. Manuf. 28 (1), 1–12. http://dx.doi.org/10.1109/TSM.2014.2364237.

Yang, X., Song, Z., King, I., Xu, Z., 2021. A survey on deep semi-supervised learning. arXiv preprint arXiv:2103.00550.

Yang, S.-B., Yu, T.-l., 2020. Pseudo-representation labeling semi-supervised learning. arXiv preprint arXiv:2006.00429.

Yu, E., Sun, J., Li, J., Chang, X., Han, X.-H., Hauptmann, A.G., 2018. Adaptive semi-supervised feature selection for cross-modal retrieval. IEEE Trans. Multimed. 21 (5), 1276–1288.

Yuan, T., Kuo, W., Bae, S.J., 2011. Detection of spatial defect patterns generated in semiconductor fabrication processes. IEEE Trans. Semicond. Manuf. 24 (3), 392–403. http://dx.doi.org/10.1109/TSM.2011.2154870.

Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L., 2019. S4l: Self-supervised semi-supervised learning. In: Proceedings of IEEE/CVF International Conference on Computer Vision. pp. 1476–1485.