SWaCo: Safe Wafer Bin Map Classification With Self-Supervised Contrastive Learning

Min Gu Kwak, Young Jae Lee, and Seoung Bum Kim

***Abstract*—Defect patterns exhibited in wafer bin maps (WBMs) can provide essential clues about critical process failures to field engineers. In modern manufacturing processes, the automatic WBM defect pattern classification is critical for yield improvement. Although it is difficult to collect sufficient labels while a lot of unlabeled data is given, most existing studies have mainly used only labeled WBM data. Moreover, the unlabeled out-of-distribution (OOD) WBMs are inevitably collected. It degrades the performance of semi-supervised models. To this end, we propose a method for safe wafer bin map classification with self-supervised contrastive learning (SWaCo) to effectively exploit the unlabeled data with OOD. We propose a loss function to utilize label information in the pre-training step to learn more suitable representations for the downstream WBM classification task. The negative labeled examples of the same class as the anchor are used for additional positive examples. Moreover, we investigate proper data augmentation for WBM self-supervised contrastive learning. To evaluate the performance and the applicability of the proposed method, experiments are conducted on a public benchmark WBM dataset, WM-811K. The results demonstrate that the proposed method achieves better classification accuracy than existing methods, especially when only a small amount of class information is given.**

***Index Terms*—Wafer bin map defect pattern classification, semiconductor manufacturing, self-supervised contrastive learning, semi-supervised learning, class distribution mismatch, out-of-distribution.**

I. Introduction

IN THE semiconductor industry, wafer fabrication is a lengthy procedure that consists of thousands of complicated processes to produce integrated circuits (IC) on each wafer. Demand for high-performance semiconductor products is increasing worldwide; accordingly, many companies are making numerous efforts to improve productivity [1]. The primary approach to increasing productivity is to reduce the size of IC chips (dies) and put more IC chips on a single wafer. As a result, the production process becomes more sophisticated. However, this made the production process difficult, and even a slight error could cause severe inline defects, eventually causing circuit failures and reducing overall yield [2].

For quality management and evaluation of wafers, manufacturing companies perform various inspections. After wafer fabrication before the packaging process, some of the wafers are randomly sampled. Then various types of electrical tests are performed to evaluate whether each IC chip meets the product specifications. An electrical die sorting (EDS) test is one of the widely used test methods, which inspects the electrical characteristics of each die of a wafer to check whether it is operating correctly. The EDS test results can be expressed as a wafer bin map (WBM). WBM portraits each die as zero if it passed the test and one if it failed. WBM can be regarded as a one-channel and two-dimensional image containing information about the size of a wafer, the position of dies, and the inspection results. Defective dies sometimes form a local pattern distinct from global random noises that appear over a wafer. Identifying the spatial and local defective patterns is highly important in the field because they are usually caused by corresponding processes [3]. A local defect pattern indicates that some systematic errors exist. Therefore, classifying WBM defect patterns can provide hints on which parts of the current operating processes have problems based on domain knowledge. It helps experienced process engineers to take appropriate maintenance in time.

Conventionally, semiconductor process engineers manually look at WBM images and classify defective patterns directly. However, modern fabs with highly complex manufacturing processes produce thousands of wafers daily. Even if an engineer samples only a small portion of the entire wafers, the amount is too large to check every WBM with the naked eye. Furthermore, the defect patterns have also become more diverse, and there are cases where it is even hard to classify into a specific category. Therefore, it is time-consuming and inconsistent for engineers to determine defect patterns manually. Many data-driven studies based on statistical learning, machine learning, and deep learning have been proposed to address these issues that occur in manual approaches [4], [5], [6], [7]. In particular, many recent studies have been built upon convolutional neural networks (CNNs) to leverage and analyze the graphical characteristics of WBM as images [8], [9], [10], [11], [12]. However, most studies assume a fully-supervised problem in which all WBMs have properly assigned classes.

In the real-world industry, the demand for wafer map analysis is incredibly high to improve yield, especially in

Manuscript received 29 January 2023; revised 13 April 2023; accepted 25 May 2023. Date of publication 29 May 2023; date of current version 4 August 2023. This work was supported in part by Brain Korea 21 FOUR; in part by the Ministry of Science and ICT (MSIT), South Korea, through the ITRC Support Program supervised by the IITP under Grant IITP-2020-0-01749; and in part by the Agency for Defense Development under Grant UI2100062D. *(Corresponding author: Seoung Bum Kim.)*

Min Gu Kwak is with the School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: mkwak35@gatech.edu).

Young Jae Lee and Seoung Bum Kim are with the School of Industrial Management Engineering, Korea University, Seoul 02841, Republic of Korea (e-mail: jae601@korea.ac.kr; sbkim1@korea.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSM.2023.3280891.

Digital Object Identifier 10.1109/TSM.2023.3280891

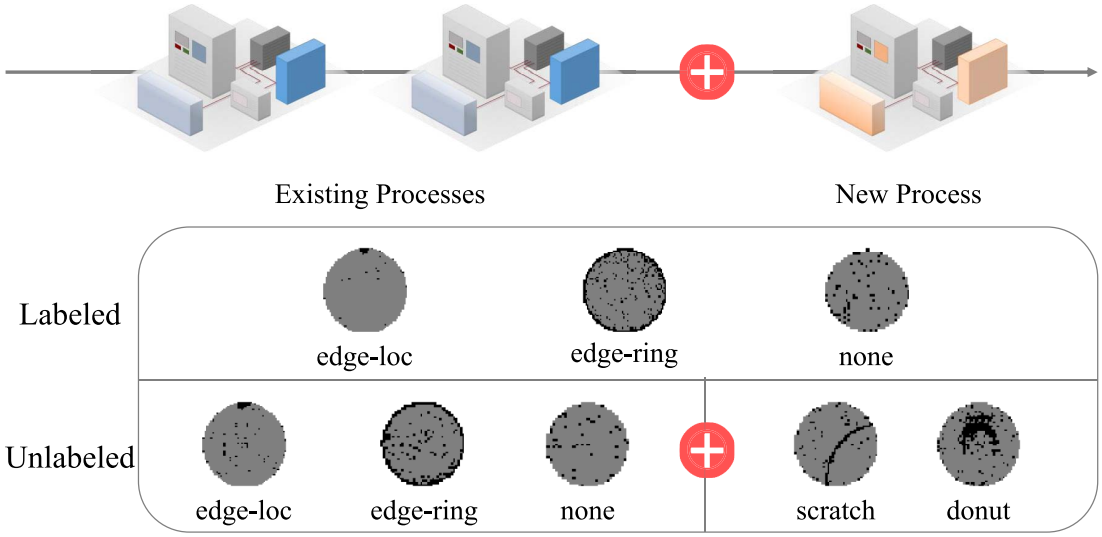

Fig. 1. Class distribution mismatch in the semiconductor manufacturing process when a new process is added. The class information that engineers have remains the same, but new defect patterns may be caused by the added process.

the process of development or early production. However, collecting enough WBM classes for model training in the initial production stage is difficult. Several methods have been proposed to address this issue in an unsupervised manner. Even if unsupervised learning could be an alternative strategy without labels, the performance is by far unsatisfactory without label guidance. Semi-supervised learning (SSL) methods have been proposed to reduce human annotation of labeling and improve model performance by utilizing unlabeled data [13]. SSL methods learn the data representation from a large amount of unlabeled data and classify new observations into pre-defined classes by taking clues from a small amount of labeled data. The methods have been widely used in many areas, including image classification, person identification, natural language processing, and health care [14], [15], [16]. However, only a few studies have been reported on wafer classification.

Furthermore, here we state a mismatch problem that might be easily found in the semi-supervised WBM classification task. There are several causes of the mismatch problem that existing studies have overlooked or assumed to be nonexistent. First, as a new fabrication process is added or an existing process becomes sophisticated, a new defective pattern that did not exist before may occur. Second, some defective patterns may be unclear to be classified into a specific category. Third, even some defective patterns may not occur so well that they have not yet been observed in the early production stages when the WBM defect pattern classification model is most necessary. These causes are likely to exist in unlabeled WBM data. Namely, the distribution of the target classes to be classified by a model and the unlabeled data distribution may differ. It is also called the class distribution mismatch problem. We could also denote the target classes as in-distribution (ID) and the classes of unlabeled data that does not present in the labeled data as out-of-distribution (OOD). Fig. 1 shows the class distribution mismatch problem in WBM when a new fabrication process is added. This problem often occurs in practice, violating the basic and strict assumption of SSL that the distributions of labeled and unlabeled data are identical. Many studies have shown that the performances of SSL methods significantly decrease when the unlabeled OOD data exist [17], [18], [19]. It is essential to consider the class distribution mismatch for WBM classification for practical applications. However, to the best of our knowledge, no study has been conducted yet.

In this paper, we propose a safe wafer bin map classification with self-supervised contrastive learning (SWaCo) using proper data augmentations to solve the class distribution mismatch problem. We hypothesize that self-supervised contrastive learning (SSCL) can provide proper initial network parameters for downstream WBM classification tasks by utilizing a large amount of unlabeled data with OOD. Unlike SSL methods, SWaCo does not assume identical distributions and does not use supervised classification loss in the pre-training stage. It leads the model to avoid the negative effect of the mismatch class included in the unlabeled data. Furthermore, unlike existing safe SSL methods addressing the class distribution mismatch problem by focusing on removing the unlabeled OOD data [17], [18], [20], the proposed method leverages the whole unlabeled data with OOD based on SSCL framework. Thus, the data representations could be effectively learned without losing information in the unlabeled data. We also propose a contrastive loss function to learn more suitable representations for WBM defect pattern classification by using class information in the pre-training step. The proposed loss function allows pre-training by mapping the ID samples with the same defective pattern class together while separating them from the OOD samples. We demonstrate the effectiveness of the proposed method with experiments on a real-world public WBM dataset, WM-811K.

II. Related Works

WBM classification is a fundamental task in the semiconductor industry because it gives meaningful clues for root-cause analysis. Because of its importance, the literature

contains a lot of studies for classifying WBM defect patterns. To date, many studies have focused on using neural networks, especially based on CNNs, which are excellent for image analysis. Convolution filters are useful for capturing local characteristics, which effectively detect locally existing defective patterns. CNN was confirmed as an effective and powerful neural network to classify defect patterns based on a synthetic WBM dataset [11]. A framework using CNNs was proposed to detect mixed-type defect patterns that are found in real wafers [10]. WBM oversampling strategy based on data augmentation was applied to CNN to solve the class imbalance problem [12]. Another oversampling method using only clear defect patterns to avoid the negative effect of unclear samples was proposed [21]. CNN classifier was applied to unsupervised embeddings of WBMS to classify different defect patterns on a real dataset [9]. Combining the predictions from a machine learning classifier based on handcrafted features and CNN outputs enhanced the classification accuracy [8]. Although the existing studies have yielded great performances in each setting, they are limited to supervised training when sufficient clean labels are available.

SSL methods perform class classification with a small amount of label data while extracting semantic data structure from a large amount of unlabeled data to improve the generalization performance. Recent deep SSL methods applying neural networks have been studied in various application areas. An expectation maximization-based deep SSL method was proposed for histopathological image segmentation [22]. A CNN network based on dilated temporal convolutions effectively estimated 3D human poses in the video when labeled data are scarce [23]. A two-stage framework based on label propagation was applied for energy consumption prediction in steel industry [24]. An autoencoder was used in a semi-supervised manner for end-to-end neural machine translation when a limited quantity of labeled parallel corpora is available [25]. A CNN with an uncertainty filter to select reliable unlabeled samples was used to detect façade defects [26].

Despite these various application cases, there are few SSL studies on WBM defect pattern classification. A deep generative SSL model based on CNN was proposed for multi-label classification by adopting multiple latent class variables [27]. A hybrid method using a ladder network and semi-supervised variational autoencoder was proposed for WBM classification [28]. An enhanced labeling method for questionable WBM samples using both unsupervised and supervised models was proposed to improve classification accuracy [29]. A semi-automatic framework supported by the process engineer's manual classification of uncertain samples was proposed [30]. However, the existing studies do not consider the scenario when OOD data exists in the unlabeled WBM data.

Recently, safe SSL methods have been proposed to address the class distribution mismatch problem. Deep safe semi-supervised learning (DS3L) imposes different weights on the unlabeled data by a be-level meta-learning optimization. It reduces the impact of the unlabeled OOD data during model training [18]. The uncertainty-aware self-distillation (UASD) dynamically filters out the unlabeled OOD data based on confidence scores calculated from labeled validation samples. The confidence scores in a moving average manner for stabilizing the filtering process [17]. An end-to-end multi-task curriculum model was proposed to simultaneously conduct classification and OOD detection. It selects the unlabeled samples by their OOD scores to calculate SSL loss [20]. OpenMatch proposed to utilize one-vs-all classifiers to learn ID representations while rejecting OOD [31]. The existing safe SSL methods are limited to natural image classification task. Moreover, they focus on discarding or minimizing the effect of the unlabeled OOD data, which might lead to the loss of common information in data.

SSCL is a research field that has recently received huge attention because of its excellent performance, especially in computer vision. It exploits the unlabeled data while reducing the labeling costs. The key concept is to make the features obtained by using different data augmentation strategies on one observation similar to each other and different from the features obtained from the other observations. It is known that general data features can be extracted without class information through the concept. With its excellence, there are many applications using SSCL. A multi-model method for ultrasound video-speech data in medical imaging was proposed to reduce expensive manual annotations [32]. The transformation of a video into a set of audio-visual objects for self-supervised learning enhanced model performances in several speech-oriented tasks [33]. The self-supervised learning model using the distinct characteristics of histopathology images that are distinguished from natural images showed promising effects in many datasets [34]. A SSCL method that exploits the spatio-temporal structure of geo-located remote sensing dataset improved classification, object detection, and semantic segmentation performances [35]. The use of SSCL methods in WBM classification has rarely been reported. Pretext-invariant representation learning (PIRL)-based model with proper data augmentation techniques for WBM was proposed [36]. However, the previous SSCL works do not take account of the class distribution mismatch problem in the wafer manufacturing process. We demonstrate that our method based on the SSCL framework can improve the classification performance under class distribution mismatch scenario by utilizing the favorable characteristic of SSCL. Our method leverages the whole unlabeled data to learn a good data representation as well as avoid the negative effect of the mismatch unlabeled samples.

III. Proposed Method

We propose SWaCo when OOD WBM exists in the unlabeled data. SWaCo is built upon MoCo [37], [38], one of the SSCL methods with prominent model performance. The key idea of the proposed method is the semi-supervised contrastive loss function that uses the labeled negative examples of an anchor's class as additional positive examples. In this section, we give a brief review of MoCo and introduce the proposed loss function for WBM defect classification with class distribution mismatch.

In SSL settings, labeled WBM data $\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^{n}$ of n samples and unlabeled WBM data $\mathcal{D}_U = \{x_j\}_{j=1}^{m}$ of m samples

are given. $\mathcal{D} = \mathcal{D}_L \cup \mathcal{D}_U$ is a total training WBM data. $x \in \mathbb{R}^{h \times w}$, $y \in \{1, 2, \ldots, C\}$, and $m \gg n$ where h, w, and C are the height, width of a wafer, and the number of defect pattern classes, respectively. A deep SSL method trains a network by leveraging both $\mathcal{D}_L$ and $\mathcal{D}_U$. SSCL model is pre-trained with $\mathcal{D}$ without any class information to capture the overall data representations. Then, it is fine-tuned with $\mathcal{D}_L$ for a specific downstream task.

In the pre-training step, MoCo aims to learn the semantic data representations through instance discrimination without any class information. Namely, it assumes that each instance belongs to its own class. It is based on the intuitive concept of making similar instances pull each other together and dissimilar instances push each other away. Primarily, given an image $x_i \in \mathcal{D}$, a stochastic data augmentation is applied to the same image twice to generate two different views: x_i^a and x_i^+, which are denoted as anchor and positive example, respectively. Similarly, K negative examples $\{x_{i,k}^-\}_{k=1}^K$ can be obtained by also applying data augmentation to a subset of images sampled from $\mathcal{D}_i = \mathcal{D} - \{x_i\}$. Namely, negative examples denote the augmented views generated from images other than x_i.

Each augmented view goes through a network f to achieve a nonlinear mapping $f : \mathbb{R}^{h \times w} \rightarrow \mathbb{R}^P$ from the data space to a P-dimensional representation space. The embedded representation vector $z = f(x)$ is L2-normalized to easily calculate the similarity between representations. A positive representation pair (z_i^a, z_i^+) is desired to be closely located because it is originated from the same source image x_i, while any negative representation pair $(z_i^a, z_{i,k}^-)$ is desired to be located far away. To meet this purpose, the contrastive loss function for the sample x_i is formulated as follows:

$$\mathcal{L}_{i,MoCo} = -\log \frac{\exp\left(z_i^a \cdot z_i^+ / \tau\right)}{\exp\left(z_i^a \cdot z_i^+ / \tau\right) + \sum_{k=1}^K \exp\left(z_i^a \cdot z_{i,k}^- / \tau\right)}, \tag{1}$$

where τ is a temperature hyperparameter for scaling. The inner dot product between the L2-normalized z representations can be regarded as cosine similarity. We can intuitively understand that $\mathcal{L}_{i,MoCo}$ is related to the probability that the anchor is classified as the positive example with the same source as itself among one positive and K negative examples. It makes MoCo perform instance discrimination in an unsupervised manner to learn the subtle semantic data structure.

It is important to note that MoCo uses two kinds of networks that share an identical architecture: key network $f_\phi(\cdot)$ and query network $f_\theta(\cdot)$, where ϕ and θ refer to the corresponding network weight parameters. The anchor representation z_i^a is obtained by the query network while the positive representation z_i^+ and negative representation $z_{i,k}^-$ are obtained by the key network. MoCo updates ϕ by a momentum update strategy of θ as follows:

$$\phi \leftarrow \lambda \cdot \phi + (1 - \lambda) \cdot \theta, \tag{2}$$

where $\lambda \in [0, 1)$ is the momentum coefficient. λ is typically set by a large value, such as 0.950 or 0.999, to slightly change the key representations. The momentum update does not change the key representation significantly and consequently makes the instance discrimination task difficult enough. It allows MoCo to generate proper and general data representations [38]. Finally, MoCo stores the past representations obtained from the key network in a memory queue. The memory queue enables MoCo to store and retrieve past representations easily during training with practicable computational costs.

After pre-training, MoCo is trained on $\mathcal{D}_L$ through the fine-tuning step. The projection layer located at the end of the network is replaced with a single layer with a softmax activation function. Namely, a softmax classifier is attached to the MoCo's backbone network. The network parameters obtained from the pre-training step are used as initial parameter values. The classification training task is performed on the labeled data, and the entire network is trained.

It is known that MoCo learns a representation in the pre-training step that can be extensively used regardless of the type of downstream task. Therefore, MoCo could be simply used for SSL. However, it is desired to obtain more suitable features for the semi-supervised classification task with unlabeled OOD WBM data. In this problem, we expect the model to map the ID samples with the same defective pattern class together while separating them from the OOD samples. To address this consideration, we propose a contrastive loss function that exploits the label information in the pre-training step. Under the SSL scenario, we can explicitly use the given limited class information to learn representations for downstream classification task. In the proposed method, the class information is paired with the representation vectors of negative examples and stored together in the memory queue. Using this information, the labeled negative examples with the same class as anchor can be used as additional positive examples.

From the perspective of training the SSCL model, selecting additional positive examples should be done very carefully. Selecting a sample with semantic characteristics different from the anchor could severely degrade the training performance. This problem is more likely to occur in the class distribution mismatch scenario. Therefore, we select samples with the same class from negative examples as additional positive examples when the class information of the anchor is given. With a representation z_m that has the same class as the anchor, the loss function is formulated as follows:

$$\mathcal{L}_{i,Class} = -\frac{I(i)}{|M(i;t)|} \log \sum_{m \in M(i;t)} \frac{\exp\left(z_i^a \cdot z_m / \tau\right)}{\exp\left(z_i^a \cdot z_i^+ / \tau\right) + \sum_{k=1}^K \exp\left(z_i^a \cdot z_{i,k}^- / \tau\right)}, \tag{3}$$

where $I(i)$ is an indicator function that is one if an x_i^a has a label, otherwise zero. $M(i; t)$ refers to the indices of negative examples having the same class as the anchor of sample i in the memory queue at iteration t. Note that the memory queue is continuously updated throughout the training iterations. Namely, z_m in the nominator of indicates the additional positive example. Fig. 2 presents an overview of the proposed loss function $\mathcal{L}_{i,Class}$.

The final loss function for SWaCo is formulated as follows:

$$\mathcal{L}_{i,SWaCo} = \alpha w(t) \mathcal{L}_{i,Class} + \mathcal{L}_{i,MoCo}, \tag{4}$$

where α is a hyperparameter for balancing two loss functions. After pre-training is complete, the proposed model is fine-tuned using $\mathcal{D}_L$, the same as vanilla MoCo.

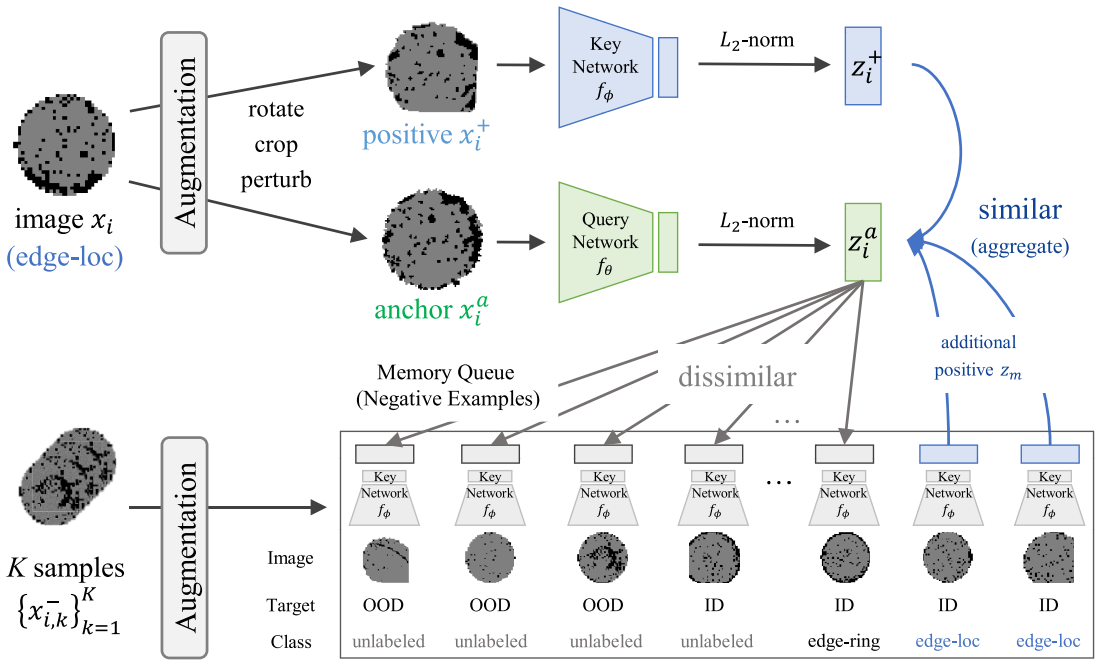

Fig. 2. Graphical overview of SWaCo. The anchor and positive example are generated by applying data augmentation twice to the same image, and are embedded as representation vectors by query and key network, respectively. Negative examples are also generated by applying data augmentation to K sampled images and go through the key network. Every representation vector is L2-normalized. Based on the class information stored in a memory queue, the negative examples with the same class as the anchor are aggregated as additional positive examples.

TABLE I
CLASS DISTRIBUTION OF LABELED DATA IN WM-811K. ID AND OOD CLASSES OF EACH SCENARIO ARE INDICATED

Class	Number of Samples	Proportion (%)	Target of Scenario		
			A	B	C
None	147,431	85.24	ID	ID	ID
Edge-Ring	9,680	5.60	ID	ID	OOD
Edge-Loc	5,189	3.00	ID	ID	OOD
Center	4,294	2.48	OOD	ID	OOD
Loc	3,593	2.08	OOD	ID	OOD
Scratch	1,193	0.69	OOD	OOD	ID
Random	866	0.50	OOD	OOD	ID
Donut	555	0.32	OOD	OOD	ID
Near-Full	149	0.09	OOD	OOD	ID

IV. EXPERIMENTS

A. Data

To demonstrate the performance of the proposed method, we conducted experiments on the public WBM dataset, WM-811K [39]. A total of 811,457 samples are obtained from real-world wafer fabrication processes. As demonstrated in Table I, only a subset of 172,950 samples has the corresponding class information, while the rest of 638,507 samples are unlabeled. We divided the labeled subset into ID and OOD subsets to examine the performance of the proposed method in the class distribution mismatch scenario. Note that we considered the frequency of each defect pattern to define the subsets. Although it is necessary to classify all defect patterns in practice, collecting labels becomes challenging for rarer defect patterns. Therefore, we primarily included high-frequency classes in an ID subset for experimental purposes.

Among the given nine classes of the labeled dataset shown in Table I, we have constructed experimental datasets using the WM-811K dataset according to three distinct scenarios for a comprehensive evaluation. It is important to note that all scenarios include *None*, a normal or white noise defect pattern, in the ID dataset. From the perspective of applying the model to the real industry, it is important to develop a model that can distinguish between normal and defective wafers and classify between the defective wafer types. In Scenario A, we defined *edge-ring*, *edge-loc*, and *none* as ID or target classes, and others as OOD classes. *Edge-ring* and *edge-loc* have local defect patterns on the edge of the wafer in common. *Edge-ring* refers to a defective pattern with defective ICs along the edge of the wafer, and *edge-loc* refers to a defective pattern with defective ICs clustered around the edge of the wafer. Moreover, they could be considered as frequently caused defect patterns in terms of their proportions greater than or equal to 3%. Scenario B extends this by additionally incorporating *center* and *loc* classes with proportions greater than or equal to 2%. Lastly, in a reverse approach, Scenario C employs *none*, *scratch*, *random*, *donut*, and *near-full* classes as ID classes, which were not designated as such in Scenarios A and B. This methodology allows for a thorough and diverse evaluation of the dataset, ensuring robust and reliable results.

We randomly split the dataset into three subsets: 80% for training, 10% for validation, and 10% for testing while preserving the class distribution. To construct the dataset for SSL under class distribution mismatch, only 10% of the labels in the training dataset were left, and the rest were discarded. Because OOD classes are not of interest, we only used the labeled ID data as $\mathcal{D}_L$. $\mathcal{D}_U$ was made with unlabeled ID and

OOD class data. The WM-811K dataset consists of WBMs with diverse spatial resolutions and most samples are not square. As training CNNs with applying data augmentations such as rotation requires a fixed square input image size, we resized WBMs to a size of 64×64 using nearest-neighbor interpolation.

B. Model Configuration and Training Hyperparameters

We compared SWaCo with the supervised classification, two SSL methods, and two safe SSL methods. Virtual adversarial training (VAT) is a well-known SSL method that enhances model robustness against local perturbations by measuring the local smoothness of conditional class distribution [40]. Stochastic weight averaging (SWA) is an SSL method that shows great classification performance. It improves model generalization over stochastic gradient descent by averaging network parameters along the training procedure [41]. DS3L [18] and UASD [17] are recently proposed safe SSL methods. DS3L utilizes an extra network updated in a meta-learning manner to control the influence of unlabeled data adaptively. UASD is a self-distillation framework to filter out the OOD data during training based on confidence scores. The models were trained for 100 epochs with a batch size of 512. The balancing coefficients for the unlabeled loss function were set to one. We optimized the network parameters with the stochastic gradient descent (SGD) optimizer with a learning rate of 0.03. The learning rate gradually decreases to zero by following a half-period cosine schedule.

For the proposed method, we followed the network structure and hyperparameters recommended in MoCo-v2 unless otherwise specified [37]. We selected a ResNet-18 as the base architecture [42] and adjusted the network to fit WBM data smaller than ImageNet [43]. We removed the max-pooling layer and replaced the first 7×7 convolution filter with a 3×3 convolution filter with a stride of one and zero padding of one. Ghost normalization was applied to mimic the batch shuffling normalization in multiple GPUs used in the original MoCo [44]. In all experiments, we split a batch into eight sub-batches to follow MoCo's eight-GPU setting. We set $\lambda = 0.999$, and 4,096 as the size of the memory queue. Because τ is a critical hyperparameter in SSCL and depends on the specific dataset, we determined the value of 0.1 by grid search as demonstrated in Section V-B. We pre-trained the model for 100 epochs with a batch size of 512. We used an SGD optimizer with a learning rate of 0.001. Starting from the initial learning rate, the learning rate gradually decreases to zero by following a half-period cosine schedule.

C. Model Evaluation

We evaluated the quality of representations obtained from the pre-trained model by using linear classification, a commonly used evaluation protocol [45], [46]. Like the fine-tuning step, the projection layer of MoCo is replaced with a softmax classifier. The parameters of the backbone network are frozen to extract the learned representations in the pre-training step and only the replaced classifier is trained using $\mathcal{D}_L$. It can be considered as training a logistic regression model with the extracted features from the pre-trained network. We trained the linear classifier for 100 epochs with an SGD optimizer with a learning rate of 30.

Eventually, we conducted a supervised fine-tuning to evaluate the model's classification performance. With the labeled data $\mathcal{D}_L$, we optimized the classifier with an SGD optimizer with a learning rate of 0.03. The learning rate gradually decreases to zero by following a half-period cosine schedule. For all experiments, we conducted ten repeated trials with different random seeds. We reported the average values of macro F1-score, the arithmetic mean of class-wise F1 scores, to reflect the class imbalance.

TABLE II
PERFORMANCE COMPARISONS OF MOCO TRAINED WITH DIFFERENT DATA AUGMENTATION UNDER SCENARIO A. LINEAR CLASSIFICATION ON THE VALIDATION DATASET IS CONDUCTED. **BOLDFACE** VALUE REPRESENTS THE BEST PERFORMANCE

Augmentation	Macro F1	Augmentation	Macro F1
rotate	72.35	rotate + crop	68.56
crop	64.49	crop + perturb	65.76
perturb	69.55	rotate + perturb	**74.55**
		rotate + crop + perturb	69.74

V. EXPERIMENTAL RESULTS

We conducted experiments with Scenario A to select optimal hyperparameters and evaluate the model performances along different proportions of labeled data. Then, we also evaluated the classification performances with Scenarios B and C to investigate the robustness of our model.

A. Data Augmentation Selection

We conducted an experiment to select the proper data augmentation for WBM classification. Most SSCL models have been studied using natural image datasets, such as ImageNet. The model performances significantly change depending on which data augmentation is applied. Some well-known data augmentations might not be suitable or even cannot be applied for WBM classification. Color jittering and grey scaling are not applicable for single-channel WBM data. Cutout might harm the class characteristics of WBM data [47]. For instance, if the circular defect pattern of center image is removed by cutout, it becomes an image with the same characteristics as *none*.

Among widely used geometric data augmentations, we compared the effect of *rotation*, *cropping*, *translation*, and *shearing*. Because *translation* and *shearing* slightly modify an image, we applied them at the same time and denoted them as *perturbation*. We also compared the compositions of the individual data augmentations. To examine the effect of the choice of data augmentation on the SSCL model, we conducted the linear classification with MoCo. Table II shows the result of linear classification on the validation dataset. Fig. 3 shows the augmented images of WBM images in ID classes.

As shown in Table II, *rotation* showed the best performance among the augmentation methods applied individually, while *cropping* showed the worst performance. We interpreted that

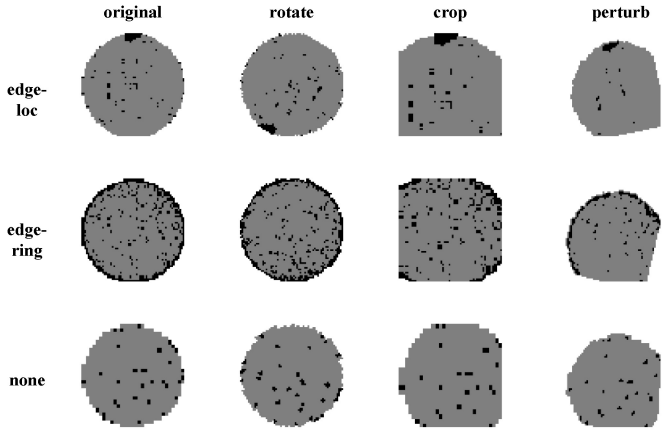

Fig. 3. Data augmentations in our work. The combinations of the data augmentations were also applied.

TABLE III
LINEAR CLASSIFICATION RESULTS OF MOCO OBTAINED OVER DIFFERENT VALUES OF τ UNDER SCENARIO A. **BOLDFACE** VALUE REPRESENTS THE BEST PERFORMANCE

τ	0.03	0.05	0.07	0.1	0.2	0.3
Macro F1	73.35	73.55	73.74	**74.55**	71.91	70.96

cropping performed the worst because, depending on the cropping scale and position, images might sometimes lose class information. For instance, if the opposite side of the defect pattern of an *edge-loc* image is cropped, it loses the defect pattern. For this reason, performance degraded when *rotation* and *cropping* were applied together. We could find that adding perturbation enhanced the model performances. As a result, using *rotation* and *perturbation* together showed the best performance, and we used this data augmentation in all remaining experiments.

B. Hyperparameter Selection

In the context of SSCL, the temperature hyperparameter τ plays a key role in enhancing the representation quality. To identify the most suitable value of τ for the WBM classification problem, we conducted a experiment by training MoCo with various τ values, including 0.03, 0.05, 0.07, 1.0, 2.0, and 3.0. Subsequently, we employed linear classification. Table III demonstrates the results on the validation dataset. The best performance of 74.55 was achieved with $\tau = 0.1$. Consequently, we fixed τ to 0.1 for all subsequent experiments.

With the fixed data augmentation and τ, we investigated the effect of hyperparameter α for pre-training by linear classification on the validation dataset. Selecting proper α is important because the balancing hyperparameter of additional loss function generally affects model performance a lot. We conducted experiments varying the values of α from 0.5 to 10. Table IV shows the results. The representation qualities obtained from the proposed method were better than the performance of 74.55% of the MoCo described in Table III, regardless of the α value. We achieved the best result of 88.93% when $\alpha = 5$. The model performance gradually increased until the α became five, but it tended to be saturated and decreased when it was greater than five. It can be interpreted that too much weight was given to grouping samples of the same class, and thus learning the overall pattern of the data was not performed effectively. With the results, we fixed α to five for all remaining experiments.

TABLE IV
LINEAR CLASSIFICATION RESULTS OF SWACO OBTAINED OVER DIFFERENT VALUES OF α UNDER SCENARIO A. **BOLDFACE** VALUE REPRESENTS THE BEST PERFORMANCE

α	0.5	1	2	3	5	7	10
Macro F1	78.86	82.21	84.44	86.26	**88.93**	88.35	87.54

TABLE V
COMPARISONS OF THE REPRESENTATION QUALITY BETWEEN MOCO AND SWACO UNDER SCENARIO A. MACRO F1-SCORES FROM THE LINEAR CLASSIFICATION ARE REPORTED. THE BEST VALUE FOR EACH LABEL PROPORTION IS IN **BOLD**

Model	Label Proportion				
	20%	10%	5%	2%	1%
MoCo	77.55	74.55	70.79	67.31	66.07
SWaCo	**89.28**	**88.93**	**87.07**	**86.35**	**83.74**

C. Representation Quality of SWaCo

With the fixed data augmentation, τ, and α, we compared the representations obtained from the proposed method with those from vanilla MoCo. Linear classification for pre-trained MoCo and SWaCo was conducted on the validation dataset. Moreover, because our method uses the class information in the pre-training step, it is important to investigate how our method scales with the proportion of the labeled data. Following the standard SSL evaluation protocol, we varied the label proportion of training data from 20% to 1%. Note that the previous experiments were conducted with a label proportion of 10%. In Table V, it can be observed that the proposed model performs better than MoCo regardless of the label proportion of the training data. Overall, the proposed model shows 15.82% higher model performance than MoCo on average. Because the model performance does not significantly degrade along the label proportion, we can extract representations suitable for the downstream task by the proposed method even when only a small amount of class information is given.

D. Classification Accuracy

Table VI presents the classification results of the comparative models and the proposed method on the testing dataset. MoCo and SWaCo were fine-tuned. Supervised refers to a model that trained with only labeled data $\mathcal{D}_L$. VAT and SWA are SSL methods, DSL and UASD are safe SSL methods, and MoCo and SWaCo are SSCL methods. Overall, the model performance is high in the order of SSCL, safe SSL, SSL, and supervised learning, regardless of label proportion. UASD showed the best performance among existing methods. Nevertheless, when MoCo was simply applied, about 1% $\sim$ 2% performance improvements were obtained.

TABLE VI
CLASSIFICATION PERFORMANCE OBTAINED OVER DIFFERENT LABEL PROPORTIONS UNDER SCENARIO A. **BOLDFACE** VALUES REPRESENT THE BEST MACRO F1-SCORE FOR EACH LABEL PROPORTION

Model	Label Proportion				
	20%	10%	5%	2%	1%
Supervised	90.95	88.98	87.65	81.02	79.09
VAT	91.02	89.36	88.04	84.12	81.54
SWA	92.01	89.73	88.51	85.16	82.03
DS3L	92.12	90.27	88.77	86.11	83.09
UASD	92.20	90.10	88.72	86.46	83.39
MoCo	92.41	91.38	90.23	88.20	85.56
SWaCo	**92.84**	**92.20**	**90.86**	**89.49**	**87.14**

TABLE VII
CLASSIFICATION PERFORMANCE OBTAINED UNDER SCENARIOS B AND C. **BOLDFACE** VALUES REPRESENT THE BEST MACRO F1-SCORE FOR SCENARIO

Model	Scenario B	Scenario C
Supervised	82.96	82.22
VAT	83.51	82.83
SWA	83.30	82.99
DS3L	83.98	84.26
UASD	84.29	84.18
MoCo	86.31	85.15
SWaCo	86.52	86.58

MoCo can effectively learn good representations even though OOD samples are included in the unlabeled data.

Moreover, SWaCo further improved the classification performances. Although the performance gaps between the proposed model and MoCo are not as significant as in the linear classification experiments because the pre-trained networks are not frozen, the proposed model yielded substantial improvements. It is important to note that SWaCo performed better by a large gap than the existing methods in challenging scenarios when only 1% or 2% of class information is available. Furthermore, in Tables V and VI, it can also be observed that the linear classification results of SWaCo are better than the existing safe SSL methods in a low class regime. For instance, when there is only 1% of class information, the linear classification of the proposed method yielded 83.74%, while UASD yielded 83.39%. Recall that the linear classification can be considered as training a logistic regression model with the extracted representations. Thus, we claim that SWaCo can be usefully applied in the wafer manufacturing process, where labeling cost is expensive.

Lastly, Table VII displays the classification performance results for Scenarios B and C. The results exhibit a performance trend similar to Scenario A, with SSCL, safe SSL, SSL, and supervised learning ranking in that order. Especially, under the Scenario C, which has the smallest sample size for ID subset classes among the scenarios, SWaCo achieved a substantial performance enhancement compared to other models. This illustrates that SWaCo consistently delivers high performance across various dataset configurations.

VI. CONCLUSION

In this paper, we proposed a method for safe wafer bin map classification with SSCL to address the class distribution mismatch problem that can be frequently encountered in WBM defect pattern classification. To the best of our knowledge, it is the first study to consider the SSL problem in the presence of OOD in the unlabeled WBM data and to use a SSCL approach for the problem. SWaCo, the proposed method, contributes to considerable gains in classification performance. We proposed a contrastive loss function to generate suitable data representations for the downstream WBM defect pattern classification tasks. In particular, the negative labeled examples of the same class as the anchor were used for additional positive examples. Furthermore, we investigated the effect of data augmentation selection for WBM defect pattern classification and found that using rotation, translation, and shearing simultaneously led the SSCL model to provide better representations. To verify the proposed method, we conducted experiments on the publicly accessible WBM dataset, WM-811K, with various scenarios. It was confirmed that the proposed method consistently achieved better performances than existing SSL and safe SSL methods, especially when labels are scarce.

Although the proposed method achieved good results, several interesting research directions exist. First, we can consider the class imbalance problem. As observed from the WM-811K dataset, the frequency of defect patterns occurring in the semiconductor industry is highly imbalanced. It is well known that the classification performance for a minority class is relatively low in an imbalanced classification problem. In product quality control, the minority defective patterns should be effectively detected. It is worth developing a method to explicitly reflect the class imbalance problem in the pre-training step of SSCL. Furthermore, the proposed method can be extended to a multi-label classification model for classifying mixed-type defect patterns. Because of dense circuits and complex wafer design, mixed-type defect patterns are easily observed in recently produced wafers. Because one WBM has several types of class information, it is necessary to design a new strategy to select the additional positive examples for the proposed loss function in the pre-training step.

REFERENCES

[1] J. Shim, S. Cho, E. Kum, and S. Jeong, "Adaptive fault detection framework for recipe transition in semiconductor manufacturing," *Comput. Ind. Eng.*, vol. 161, Nov. 2021, Art. no. 107632.
[2] T. Yuan, W. Kuo, and S. J. Bae, "Detection of spatial defect patterns generated in semiconductor fabrication processes," *IEEE Trans. Semicond. Manuf.*, vol. 24, no. 3, pp. 392–403, Aug. 2011.
[3] M. H. Hansen, V. N. Nair, and D. J. Friedman, "Monitoring wafer map data from integrated circuit fabrication processes for spatially clustered defects," *Technometrics*, vol. 39, no. 3, pp. 241–253, 1997.
[4] F. Adly, P. D. Yoo, S. Muhaidat, Y. Al-Hammadi, U. Lee, and M. Ismail, "Randomized general regression network for identification of defect patterns in semiconductor wafer maps," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 2, pp. 145–152, May 2015.
[5] C.-F. Chien, S.-C. Hsu, and Y.-J. Chen, "A system for online detection and classification of wafer bin map defect patterns for manufacturing intelligence," *Int. J. Prod. Res.*, vol. 51, no. 8, pp. 2324–2338, 2013.

[6] C.-S. Liao, T.-J. Hsieh, Y.-S. Huang, and C.-F. Chien, "Similarity searching for defective wafer bin maps in semiconductor manufacturing," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 953–960, Jul. 2014.

[7] M. Piao, C. H. Jin, J. Y. Lee, and J.-Y. Byun, "Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 250–257, May 2018.

[8] H. Kang and S. Kang, "A stacking ensemble classifier with handcrafted and convolutional features for wafer map pattern classification," *Comput. Ind.*, vol. 129, Aug. 2021, Art. no. 103450.

[9] J. Kim, H. Kim, J. Park, K. Mo, and P. Kang, "Bin2Vec: A better wafer bin map coloring scheme for comprehensible visualization and effective bad wafer classification," *Appl. Sci.*, vol. 9, no. 3, p. 597, 2019.

[10] K. Kyeong and H. Kim, "Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 3, pp. 395–402, Aug. 2018.

[11] T. Nakazawa and D. V. Kulkarni, "Wafer map defect pattern classification and image retrieval using convolutional neural network," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 309–314, May 2018.

[12] M. Saqlain, Q. Abbas, and J. Y. Lee, "A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 3, pp. 436–444, Aug. 2020.

[13] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (Chapelle, O. et al., Eds.; 2006) [book reviews]," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, p. 542, Mar. 2009.

[14] M.-F. Balcan et al., "Person identification in Webcam images: An application of semi-supervised learning," in *Proc. Workshop Learn. Partially Classified Training Data*, vol. 2, 2005, p. 6.

[15] D. Garrette, J. Mielens, and J. Baldridge, "Real-world semi-supervised learning of POS-taggers for low-resource languages," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguist. (Long Papers)*, vol. 1, 2013, pp. 583–592.

[16] D. Sen, A. Aghazadeh, A. Mousavi, S. Nagarajaiah, R. Baraniuk, and A. Dabak, "Data-driven semi-supervised and supervised learning algorithms for health monitoring of pipes," *Mech. Syst. Signal Process.*, vol. 131, pp. 524–537, 2019.

[17] Y. Chen, X. Zhu, W. Li, and S. Gong, "Semi-supervised learning under class distribution mismatch," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 3569–3576.

[18] L.-Z. Guo, Z.-Y. Zhang, Y. Jiang, Y.-F. Li, and Z.-H. Zhou, "Safe deep semi-supervised learning for unseen-class unlabeled data," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3897–3906.

[19] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. J. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–12.

[20] Q. Yu, D. Ikami, G. Irie, and K. Aizawa, "Multi-task curriculum framework for open-set semi-supervised learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 438–454.

[21] E.-S. Kim, S.-H. Choi, D.-H. Lee, K.-J. Kim, Y.-M. Bae, and Y.-C. Oh, "An oversampling method for wafer map defect pattern classification considering small and imbalanced data," *Comput. Ind. Eng.*, vol. 162, Dec. 2021, Art. no. 107767.

[22] J. Li et al., "An EM-based semi-supervised deep learning approach for semantic segmentation of histopathological images from radical prostatectomies," *Comput. Med. Imag. Graph.*, vol. 69, pp. 125–133, Nov. 2018.

[23] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3D human pose estimation in video with temporal convolutions and semi-supervised training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 7745–7754.

[24] C. Chen, Y. Liu, M. Kumar, J. Qin, and Y. Ren, "Energy consumption modelling using deep learning embedded semi-supervised learning," *Comput. Ind. Eng.*, vol. 135, pp. 757–765, Sep. 2019.

[25] Y. Cheng, "Semi-supervised learning for neural machine translation," in *Joint Training for Neural Machine Translation*. Singapore: Springer, 2019, pp. 25–40.

[26] J. Guo, Q. Wang, and Y. Li, "Semi-supervised learning based on convolutional neural network and uncertainty filter for façade defects classification," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 36, no. 3, pp. 302–317, 2021.

[27] H. Lee and H. Kim, "Semi-supervised multi-label learning for classification of wafer bin maps with mixed-type defect patterns," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 4, pp. 653–662, Nov. 2020.

[28] Y. Kong and D. Ni, "A semi-supervised and incremental modeling framework for wafer map classification," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 1, pp. 62–71, Feb. 2020.

[29] K. S.-M. Li et al., "Wafer defect pattern labeling and recognition using semi-supervised learning," *IEEE Trans. Semicond. Manuf.*, vol. 35, no. 2, pp. 291–299, May 2022.

[30] S. Yoon and S. Kang, "Semi-automatic wafer map pattern classification with convolutional neural networks," *Comput. Ind. Eng.*, vol. 166, Apr. 2022, Art. no. 107977.

[31] K. Saito, D. Kim, and K. Saenko, "Openmatch: Open-set semi-supervised learning with open-set consistency regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 25956–25967.

[32] J. Jiao, Y. Cai, M. Alsharid, L. Drukker, A. T. Papageorghiou, and J. A. Noble, "Self-supervised contrastive video-speech representation learning for ultrasound," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2020, pp. 534–543.

[33] T. Afouras, A. Owens, J. S. Chung, and A. Zisserman, "Self-supervised learning of audio-visual objects from video," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 208–224.

[34] O. Ciga, T. Xu, and A. L. Martel, "Self supervised contrastive learning for digital histopathology," *Mach. Learn. Appl.*, vol. 7, Mar. 2022, Art. no. 100198.

[35] K. Ayush et al., "Geography-aware self-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10181–10190.

[36] H. Kahng and S. B. Kim, "Self-supervised representation learning for wafer bin map defect pattern classification," *IEEE Trans. Semicond. Manuf.*, vol. 34, no. 1, pp. 74–86, Feb. 2021.

[37] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[38] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 9726–9735.

[39] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 1–12, Feb. 2015.

[40] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[41] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson, "There are many consistent explanations of unlabeled data: Why you should average," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–22.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 2009, pp. 248–255.

[44] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: Closing the generalization gap in large batch training of neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[45] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 9912–9924.

[46] Z. Wu, Y. Xiong, X. Y. Stella, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 3733–3742.

[47] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.