SupConWI-RL: wafer inspection with reinforcement learning enhanced by supervised contrastive learning

Aleksandr Dekhovich
Delft University of Technology
a.dekhovich@tudelft.nl

Oleg Soloviev
Delft University of Technology, Flexible Optical B.V.
o.a.soloviev@tudelft.nl

2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) | 979-8-3315-8988-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICCVW69036.2025.00149

Abstract

Monitoring manufacturing processes plays an important role in chip production. Current state-of-the-art approaches use the entire surface to classify defects with CNN- or Transformer-based models, resulting in considerable measurement costs. Therefore, new advanced techniques are required to reduce the cost of inspection. In this work, we advocate for the reinforcement learning-based feedback loop with a classifier trained with supervised contrastive loss. In contrast to previous works in this manner, our approach is not limited to only one type of defect but can identify multiple defects on one wafer. We tested our algorithm on the publicly available WM-811k and MixedWM38 datasets, showing a significant reduction in scanning time compared to CNN-based approaches while maintaining similar accuracy. We demonstrate the reduction of up to 40% *in costs associated with wafer scanning in defect classification tasks, even if multiple defects are on the surface. Moreover, we demonstrate that in the multi-defect scenario, the trained model can be directly used to detect outliers, requiring only about* 12.5% *of the surface to find at least one type of defect.*

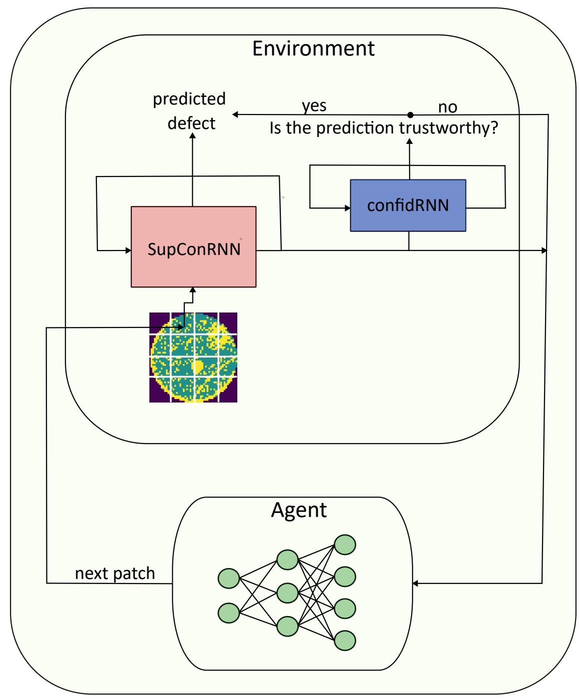


Figure 1. The overview of the proposed SupConWI-RL: SupConRNN scans a new patch and makes a prediction, while confidRNN estimates the trustworthiness of this prediction. If confidRNN is not certain, the next patch is selected by the agent.

1. Introduction

Defect inspection is a vital part of semiconductor wafer surface monitoring [33]. Early approaches rely on handcrafted features and prior domain knowledge [9, 28]. Today, deep learning has become a powerful tool for finding anomalies [30] and classification of defects in visual data [37]. However, it is often necessary to also know the type of anomaly in the real manufacturing process. Hence, multiple visual recognition algorithms have been developed over the decade in different real-world scenarios to improve surface inspection approaches using few-shot learning [21], continual learning [7] and unsupervised learning [35]. However, these approaches do not aim to reduce the costs associated with scanning and measuring the surface, which means that the full surface should be scanned to predict the defect type.

Closed-loop active model diagnosis (CLAMD) [29] has been recently proposed for an active visual inspection to minimize the number of measurements. Yet, it has scalability issues, increasing the parameters being learned as the image size increases. Also, CLAMD was validated on small images of hand-written digits and not on the images of real-world wafers. Yen *et al.* [43] proposed an RL-based sampling technique from the k-space for the pathology classification problem. Meanwhile, Sequential wafer inspection (SWI) [8] proposes a feedback loop approach with reinforcement learning for wafer defect classification. The

2473-9944/25/$31.00 ©2025 IEEE
DOI 10.1109/ICCVW69036.2025.00149

algorithm shows comparable results with CNNs, requiring fewer regions for scanning. The main drawback of SWI is its limitations to only a single-defect case, which is unrealistic to guarantee in industrial applications.

Thus, we propose SupConWI-RL, a feedback loop approach for wafer map inspection suitable for the multi-defect case where the surface can contain several defects. The algorithm employs a reinforcement learning paradigm for active decision-making and a classifier trained in a contrastive learning manner for better recognition capabilities. Also, the confidence estimator signals the algorithm when to stop scanning. Fig. 1 shows the overview of the proposed SupConWI-RL. Our contribution can be summarized as follows:

- We proposed a novel wafer inspection approach that aims to minimize the number of scanning steps, in contrast to other state-of-the-art ones in wafer map failure recognition. The algorithm is suitable for datasets with multiple defects on a wafer, which distinguishes it from previous work on this topic. We demonstrate a reduction of $37-40\%$ in scanning time compared to a full wafer scan on two publicly available datasets.
- Our trained model can be used in anomaly detection mode without any retraining, detecting at least one type of defect present on the wafer. The algorithm needs only around 12.5% on average to decide whether a wafer is normal or defective with 100% accuracy.
- We made our code implementation publicly available on GitHub so that one can reproduce our results for comparison. The link is `https://github.com/adekhovich/SupConWI-RL`.

The rest of the article is organized as follows: Section 2 gives an overview of the works related to the topic, Section 3 explains the proposed method, Section 4 shows the results and provides the comparison with the alternative approaches, Section 5 explores the necessity of every component of the algorithm and discusses its prons and cons, Section 6 concludes the work.

2. Related work

2.1. Wafer defect recognition

Deep learning models have become popular in many applications, including wafer map defect recognition. However, most of them rely on convolutional neural networks (CNNs) to identify a failure pattern [3, 22, 27, 36]. Some approaches generate additional training data using generative models, e.g., VAEs [39], GANs [18] and diffusion models [25], or using sophisticated data augmentation techniques [19] to increase the learning capabilities of conventional CNN models. Huang *et al*. [17] utilize the Focal loss to mitigate the class imbalance issue, which is common in real-world scenarios.

It is also important to mention that some methods are specialized for the single-defect case [45], while the other methods are validated in the multi-defect scenario [4]. Open databases such as WM-811k [42] and MixedWM38 [38] facilitate the development of deep learning algorithms because accessing real-world wafers is expensive and can be associated with data privacy issues.

2.2. Sparse measuremnets acquisition

As stated in Section 1, our goal is to reduce scanning time during the inspection/metrology process without sacrificing accuracy, i.e. to increase system throughput. This problem is closely related to the work on MRI data acquisition techniques using reinforcement learning. These works apply reinforcement learning to find sparse measurements that best reconstruct the MRI [2, 31, 44].

SWI [8] solves the wafer map failure recognition problem with a feedback loop, where a controller is present as an RL agent that collects sparse observations of the wafer surface. The approach consists of three core blocks: a pre-trained GRU, a confidence estimator, and the next observation selector. However, this approach is limited to the single-defect case. Thus, the development of more general approaches is required.

In the reinforcement learning framework [34], an agent interacts with the environment defined as the Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, R, \mu, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition distribution, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\mu : \mathcal{S} \to [0, 1]$ is the initial state distribution and $\gamma \in (0, 1]$ is the discount factor. The goal is to find an optimal policy $\pi : \mathcal{A} \times \mathcal{S} \to [0, 1]$ that maximizes the expected return $\mathbb{E}_\pi \left[\sum_t \gamma^t R(s_t, a_t) \right]$ by finding the optimal policy $\pi(a|s)$, where $s_0 \sim \mu(\cdot)$ is the initial state, $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$. Typically, the approaches that derive the optimal policy π are divided into the value-based methods (e.g., DQN [26]), and policy gradient ones (e.g., REINFORCE [40], PPO [32]).

2.3. Contrastive learning

In computer vision, contrastive learning aims to learn invariant features of the images and encode them into a space where embeddings of similar instances are close together while dissimilar instances are put apart. We are interested in supervised contrastive learning (SCL) [20] because we have access to the defect labels and can use this information for better embedding separation. Bae and Kang [1] proposed a novel supervised contrastive loss term that improves the wafer defect recognition capabilities of deep CNN models:

$$\mathcal{L}_{\text{con}}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{y}}_i; \tilde{\mathcal{B}}) = -\frac{1}{\sum_{j=1}^{2M} \mathbf{1}(i \neq j) w_{i,j}} \times$$
$$\times \log\left(\sum_{j=1}^{2M} \frac{\mathbf{1}(i \neq j) w_{i,j} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2M} \mathbf{1}(i \neq k) \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}\right), \quad (1)$$

where τ is a temperature hyperparameter, sim is the cosine similarity, $\mathbf{1}(\cdot)$ is an indicator function, and $w_{i,j}$ are defined by

$$w_{i,j} = \begin{cases} 1, & \text{if } \tilde{\mathbf{y}}_i = \tilde{\mathbf{y}}_j = 0; \\ \frac{\tilde{\mathbf{y}}_i \cdot \tilde{\mathbf{y}}_j}{||\tilde{\mathbf{y}}_i||_1 + ||\tilde{\mathbf{y}}_j||_1 - \tilde{\mathbf{y}}_i \cdot \tilde{\mathbf{y}}_j}, & \text{otherwise.} \end{cases}$$

The procedure for SCL for the network with feature extractor F and projection head G on the data minibatch $\mathcal{B} = \{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^{M}$ is summarized as follows:

- sample two data augmentation strategies $\mathcal{T}_1$ and $\mathcal{T}_2$;
- apply $\mathcal{T}_1$ and $\mathcal{T}_2$ to $\mathbf{X}_i$: $\tilde{\mathbf{X}}_i = \mathcal{T}_1(\mathbf{X}_i)$, $\tilde{\mathbf{X}}_{2i} = \mathcal{T}_2(\mathbf{X}_i)$, $i = 1, 2, \ldots, M$;
- collect the data into the augmented mini-batch $\tilde{\mathcal{B}} = \{(\tilde{\mathbf{X}}_i, \mathbf{y}_i)\}_{i=1}^{2M}$;
- compute $\mathbf{z}_i = G(F(\tilde{\mathbf{X}}_i)),\ i = 1, 2, \ldots, 2M$.
- compute $\mathcal{L}_{\text{con}}$, e.g., using (1).

2.4. Confidence estimation

Uncertainty estimation is an important component of reliable prediction for manufacturing processes. Bayesian approaches [41] estimate the mean and standard deviation of the neural network output. However, these methods struggle from the computational complexity of the marginal probability. Additionally, these statistics do not provide a clear understanding of when we can trust the prediction without additional thresholds, which are hyperparameters.

To mitigate the computational aspect, MC dropout [11] and Deep Ensemble [23] were proposed to approximate Bayesian inference. MC dropout uses multiple inferences with the dropout and collects statistics from this sampling. Deep Ensemble trains multiple models of the same architecture but with different parameter initialization, averaging their predictions. However, it is not easy to interpret the collected statistics in the context of a classification task to decide whether the prediction is reliable or not.

The selective classifier [12] accepts the prediction of whether a certain criterion is met and rejects it otherwise. The maximum class probability (MCP) [15] approach proposes a simple baseline that uses the output of the softmax layer as the probability that an object belongs to a certain class. However, deep learning models tend to overestimate their prediction [23], making the softmax outputs unreliable. True class probability (TCP) [6] uses an auxiliary neural network that learns the output probability of the true class.

3. Methodology

Our goal is to determine the type of defect from the sparse observations, that we call patches, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T \in \mathbb{R}^{n_1 \times n_2}$ obtained from the wafer surface $\mathbf{X} \in \mathbb{R}^{N_1 \times N_2}$, where $T \times n_1 \times n_2 = N_1 \times N_2$, meaning that these patches do not overlap. Moreover, the aim is to predict the defect using the smallest number of patches possible; the optimal number of scanning steps is defined as t^*. We follow the same paradigm that was used for MRI reconstruction [2] and SWI [8] for the single defect wafer classification. We have a pretrained model, a classifier, to make a prediction, while an RL-based controller collects observations in an optimal order to minimize their number. An additional model indicates when the scanning should stop, which means that the controller has provided the classifier with enough information for it to make a reliable prediction.

3.1. Classifier

As a classifier, we need a model that can process information sequentially at times $t = 1, 2, \ldots, T$ and make predictions $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2 \ldots, \hat{\mathbf{y}}_T$. This model is trained in random sequences of patches, receiving a patch $\mathbf{x}_{p_t}$ and its encoded position $\mathbf{e}_t = \text{Embedding}(p_t)$ at time t. For this reason, we use a recurrent neural network (RNN) [10], which means $\hat{\mathbf{y}}_t = \text{RNN}([\mathbf{x}_{p_t}, \mathbf{e}_t]; \theta_{\text{cls}})$, where θ_{cls} denotes the trainable parameters. To pretrain this RNN, we use the loss $\mathcal{L}$ that has the form

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \alpha \mathcal{L}_{\text{con}}, \quad (2)$$

where $\mathcal{L}_{\text{cls}}$ is the classification term and $\mathcal{L}_{\text{con}}$ is the supervised contrastive loss term defined in (1). Therefore, we call it SupConRNN, which can be with both GRU [5] and LSTM [16] architectures. Classification loss term $\mathcal{L}_{\text{cls}}$ depends on the type of labels that we have. In the multi-class classification case, this loss term is the categorical cross-entropy function. In the multi-label case with K single-defects, our target is $\mathbf{y} = (y_1, y_2, \ldots, y_K) \in \{0, 1\}^K$, where $y_k = 1$ if defect k is present in the wafer, and $y_k = 0$ otherwise. Thus, the loss term is the sum of the element-wise binary cross-entropy terms for every defect type. This means that for the WM-811k dataset, we use the categorical cross-entropy, while for MixedWM38, we use the sum of binary cross-entropy losses.

3.2. Confidence estimation

As we discussed previously, we need a simple rule that determines when the algorithm should stop collecting new patches. So, we use an RNN-based confidnet, a confidRNN, that predicts confidence after each timestep t. As an input, it takes the last hidden state $\mathbf{z}_t^{(L)}$ of SupConRNN, so $\hat{\mathbf{c}}_t = \text{confidRNN}(\mathbf{z}_t^{(L)}; \theta_{\text{conf}}) \in \mathbb{R}^K$. It should be noted that for the multi-defect case, our targets are one-hot encoded, $\mathbf{y} \in \{0, 1\}^K$, so this confidRNN should be certain in all components of the output. This means that it must be certain in prediction about the presence or absence of each type of defect.

For the single-label case, *i.e.* WM-811k dataset, we use TCP [6], while for MixedWM38 dataset, we modify this idea for the multi-label scenario. We define the confidence $\mathbf{c}_t = (c_{t1}, c_{t2}, \ldots, c_{tK})$ at time step t for each component as follows: $c_{tk} = y_k \sigma_{tk} + (1 - y_k)(1 - \sigma_{tk})$, where y_k is an element k in the label $\mathbf{y}$, and σ_{tk} is the output of the SupConRNN (after applying sigmoid activation). Then, the parameters θ_{conf} of confidRNN are trained minimizing weighted MSE loss $\mathcal{L}_{\text{mse}} = \sum_{t=1}^{T} \sum_{k=1}^{K} \beta(c_{tk} - \hat{c}_{tk})^2$, where $\beta = \begin{cases} \beta_0, & y_k \neq \hat{y}_{tk} \\ 1, & y_k = \hat{y}_{tk} \end{cases}$ with some hyperparameter β_0 that penalizes more the componts that were predicted wrongly.

Let us assume that $c_{tk} \geq 0.5$, then we have two possibilities:
- if true label $y_k = 1$, then $c_{tk} = \sigma_{tk} \geq 0.5 \Rightarrow \hat{y}_{tk} = 1$;
- if true label $y_k = 0$, then $c_{tk} = 1 - \sigma_{tk} \geq 0.5 \Rightarrow \sigma_{tk} \leq 0.5 \Rightarrow \hat{y}_{tk} = 0$.

Overall, if $c_k \geq 0.5$, i.e., the confidRNN signal that it is confident in SupConRNN prediction, this implies that $y_k = \hat{y}_{tk}$. On the other hand, if the confidRNN is still not certain in the prediction at the current iteration, i.e., $c_{tk} < 0.5$, we again have two possibilities:
- if true label $y_k = 1$, then $c_{tk} = \sigma_k < 0.5 \Rightarrow \hat{y}_{tk} = 0$;
- if true label $y_k = 0$, then $c_{tk} = 1 - \sigma_{tk} < 0.5 \Rightarrow \sigma_{tk} > 0.5 \Rightarrow \hat{y}_{tk} = 1$.

As a result, $c_{tk} < 0.5 \Rightarrow y_k \neq \hat{y}_{tk}$. Combining two parts together, we obtain $c_{tk} \geq 0.5 \Leftrightarrow y_k = \hat{y}_{tk}$. Thus, if we have a well-trained confidRNN that approximates c_{tk}, we have the stopping criterion for the multi-label case.

3.3. Controller

The next step is designing the controller that generates the optimal sequence of patches so that the SupConRNN requires the smallest number of measurements possible. We use the RL approach with the policy gradient method, where a neural network π_θ with trainable parameters θ takes the current state of the environment $\mathbf{s}_t$ as input and produces the probability distribution π_t over available patches as input, i.e., $\pi_t = \pi_\theta(\mathbf{s}_t)$. The state $\mathbf{s}_t = [\mathbf{z}_t^{(L)}, \mathbf{m}_t]$, where $\mathbf{z}_t^{(L)}$ is the normalized hidden state of the SupConRNN and $\mathbf{m}_t \in \{0, 1\}^T$ is a mask for the available patches. During training, we sample the next action a_t from the probability distribution π_t.

We construct a reward that considers how sure the confidRNN is about the next observed patch at time step t. Thus the reward is defined using $\hat{\mathbf{c}}_t = (\hat{c}_{t1}, \hat{c}_{t2}, \ldots, \hat{c}_{tK})$:

$$r_t = \begin{cases} 1, & \text{if } \hat{c}_{tk} \geq 0.5 \quad \forall k = 1, 2, \ldots, K, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

This way, we ensure that the controller is rewarded for selecting patches that contribute to identifying the presence of all defects. Then the total reward is defined as $G = \sum_{t=0}^{T-1} \gamma^t r_{t+1}$ that we optimize with PPO [32] algorithm.

We want to emphasize that once the SupConWI-RL is trained for defect classification, it can also be used for anomaly detection. The difference between the problems is that we need to find *all* defects during classification, while anomaly detection requires finding only one failure pattern that already signals that the surface is defective. Therefore, we have two stopping criteria at time step t for scanning during inference:
- *Classification criterion*: $\hat{c}_{tk} \geq 0.5 \quad \forall k = 1, 2, \ldots, K$;
- *Anomaly criterion*: ($\exists k : \hat{c}_{tk} \geq 0.5$ and $\hat{y}_{tk} = 1$) or ($\forall k = 1, 2, \ldots, K \quad \hat{c}_{tk} \geq 0.5$ and $\hat{y}_{tk} = 0$).

In the classification case, the criterion finds the first time step when confidRNN is certain about *all* components of the prediction $\hat{\mathbf{y}}_t$. In the anomaly detection case, we should be sure of the presence of *at least* one type of defect or the absence of *all* types. Thus, one of these conditions should be satisfied at time t.

3.4. Hybrid approach & Ensemble

As mentioned in [8], some wafers may still require full scanning because the confidence estimator is not sure of all predictions $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_T$. However, that means that all patches are collected, and we can use CNN for prediction without any increase in computational resources spent on measurements. Following [8], we also call this a *hybrid* approach. Fig. 2 illustrates this idea.

In addition, it has been observed that using ensembles improves the recognition capabilities of deep learning models [13]. Thus, we can use ensembles of M models for SupConRNN, confidGRU and policy networks as follows: $\hat{\mathbf{y}}_t = \frac{1}{M} \sum_{i=1}^{M} \hat{\mathbf{y}}_t^{(i)}$, $\mathbf{c}_t = \frac{1}{M} \sum_{j=1}^{M} \mathbf{c}_t^{(j)}$ and $\pi_t = \frac{1}{M} \sum_{l=1}^{M} \pi_t^{(l)}$. We call **Single SupConWI-RL** the model that consists of only one SupConRNN, confidRNN and policy network, and with **Ensemble SupConWI-RL**, we refer to the ensemble of M models. Algorithm 1 summarizes the approach.

4. Experiments

4.1. Setup

Dataset To validate our approach in a multi-defect case, we use the MixedWM38 dataset that contains wafers from 38 categories and all images have size 52×52. Each category contains $0, 1, 2, 3$ or 4 defect types out of $K = 8$ possible ones, and, therefore, every label is encoded in a binary vector of size 8. We split this dataset into three datasets $\mathcal{D}_1, \mathcal{D}_2$ and $\mathcal{D}_3$ in proportions 3:1:1 respectively. The dataset $\mathcal{D}_1$ is used to train the classifier, while we employ $\mathcal{D}_2$ to train the confidence estimator and the policy net-

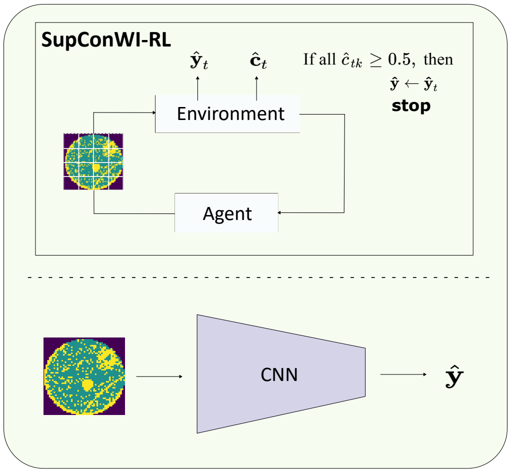


Figure 2. Hybrid approach: if SupConWI-RL is not certain after completing the full scan, the wafer is classified with a CNN model

Algorithm 1 Inference

Input: wafer map $\mathbf{X}$, the number of models in the ensemble M, stopping *criterion*.
Output: predicted defect type $\hat{y}$

Initialization: $\hat{\mathbf{y}} \leftarrow \emptyset$, $\mathbf{m}_0 \leftarrow (1, 1, \ldots, 1)$, $\mathbf{z}_0^{(L)^{(i)}} \leftarrow (0, 0, \ldots, 0)$, $i = 1, 2 \ldots, M$.

first statement
for $t = 0$ to $T - 1$ **do**
  for $i = 1$ to M **do**
    $\mathbf{s}_t^{(i)} \leftarrow [\mathbf{z}_t^{(L)}, \mathbf{m}_t]$
    $\pi_t^{(i)} \leftarrow \pi\big(\mathbf{s}_t^{(i)}; \theta_{\text{pol}}^{(i)}\big)$
  end for
  $\pi_t \leftarrow \frac{1}{M} \sum_{i=1}^{M} \pi_t^{(i)}$
  $a_t \leftarrow$ argmax π_t
  for $i = 1$ to M **do**
    $\hat{\mathbf{y}}_{t+1}^{(i)}, \mathbf{z}_{t+1}^{(L)^{(i)}} \leftarrow \text{SupConRNN}\big([\mathbf{x}_{a_t}, a_t]; \theta_{\text{cls}}^{(i)}\big)$
    $\hat{\mathbf{c}}_{t+1} \leftarrow \text{confidRNN}\big(\mathbf{z}_{t+1}^{(L)}; \theta_{\text{conf}}^{(i)}\big)$
  end for
  $\hat{\mathbf{c}}_{t+1} \leftarrow \frac{1}{M} \sum_{i=1}^{M} \hat{\mathbf{c}}_{t+1}^{(i)}$
  $\hat{\mathbf{y}}_{t+1} \leftarrow \frac{1}{M} \sum_{i=1}^{M} \hat{\mathbf{y}}_{t+1}^{(i)}$
  if *criterion* == True **then**
    $\hat{\mathbf{y}} \leftarrow \hat{\mathbf{y}}_{t+1}$
  end if
end for
if $\hat{\mathbf{y}} == \emptyset$ **then**
  $\hat{\mathbf{y}} \leftarrow \text{CNN}(\mathbf{X})$
end if
return $\hat{\mathbf{y}}$

work; the dataset $\mathcal{D}_3$ is a testing dataset. We split every image into $T = 169$ patches of size 4×4 each.

For completeness, we also validate SupConWI-RL on the WM-811k dataset to compare it with SWI [8] and show the generality of the proposed method. Similarly, we denote the training dataset as $\mathcal{D}_1$ and we split the testing dataset into $\mathcal{D}_2$ and $\mathcal{D}_3$ in the proportion $1 : 1$ and use the same way as for the MixedWM38 dataset. Each image was resized to 64×64 and divided into $T = 64$ patches of size 8×8.

Models For WM-811k dataset, we use 2 GRU cells with 256 neurons in the hidden state. In the case of MixedWM-38, the architecture of SupConLSTM contains 3 LSTM cells with Layer Normalization [24] after each of them. The hidden size is 256, and the projection size for the contrastive loss term is 128. Following the original work for the SCL loss term for the MixedWM38 dataset [1], we set $\alpha = 0.1$. The model is then trained for 500 epochs with a learning rate 10^{-3} and a weight decay equal to 10^{-5}. The confidence estimator confidLSTM consists of 2 LSTM cells with a hidden size equal to 256. We train it with the learning rate 10^{-3} and the weight decay 10^{-5} for 300 epochs. The RL controller is trained with the PPO algorithm. The policy and critic networks are multilayer perceptrons with 2 hidden layers of size 400 neurons. This network is also trained with the learning rate 10^{-3} and a weight decay equal to 10^{-5} during 100 iterations. For both datasets, the size of the ensemble is $M = 5$.

Baselines We compare our work with CNN-based approaches, including conventional CNN, CNN with SCL [1]. For the CNN model, we use ResNet-34 [14] for any baseline. In the WM-811k dataset, our main goal is to outperform the SWI approach, showing the advantage of supervised contrastive learning. It is important to state that the proposed model is compatible with any other CNN-based method in the hybrid scenario.

Metrics To evaluate our approach and compare it with others, we use three evaluation criteria:

- **Accuracy** $= \frac{\#\{i\,:\,\mathbf{y}_i = \hat{\mathbf{y}}_i,\ i=1,2,\ldots,N\}}{N} \times 100\%$ that shows the percentage of correctly predicted defects, which is an intuitively understandable indicator,
- **F1 score** $= (2 \times Recall \times Precision)/(Recall + Precision)$, which is well-suited for unbalanced datasets,
- **Number of scanned patches** t^*, which is our main target,

where N is the dataset size, $Recall = TP/(TP + FN)$, $Precision = TP/(TP + FP)$, and TP, TN, FN, and FP are the numbers for the true positive, true negative, false negative and false positive predictions, respectively.

4.2. Classification results

We start by showing the classification results. Table 1 and Table 2 compare our **Ensemble SupConWI-RL** with other

Table 1. Comparison of **Ensemble SupConWI-RL** with CNN-based methods and SWI on WM-811k dataset in terms of accuracy, F1 score and the number of scanned patches t^*

Model	Acc. (%)	F1 score	t^*
ResNet-34 [14]	85.28	0.841	$64/64$
ResNet-34 + Ensemble SWI [8]	85.38	**0.844**	$40.7/64$
ResNet-34 + **Ensemble SupConWI-RL** (ours)	**85.49**	0.838	$\mathbf{38.0}/64$
ResNet-34 w/ SCL [1]	85.52	0.848	$64/64$
ResNet-34 w/ SCL + Ensemble SWI [8]	85.41	**0.849**	$40.7/64$
SupConResNet-34 + **Ensemble SupConWI-RL** (ours)	**85.69**	0.844	$\mathbf{38.0}/64$

Table 2. Comparison of **Ensemble SupConWI-RL** with CNN-based methods on MixedWM-38 dataset in terms of accuracy, F1 score and the number of scanned patches t^*

Model	Acc. (%)	F1 score	t^*
ResNet-34 [14]	98.79	0.995	$169/169$
ResNet-34 + **Ensemble SupConWI-RL** (ours)	98.45	0.995	$\mathbf{102.1}/169$
ResNet-34 w/ SCL [1]	98.77	0.994	$169/169$
ResNet-34 w/ SCL + **Ensemble SupConWI-RL** (ours)	98.44	0.995	$\mathbf{102.1}/169$

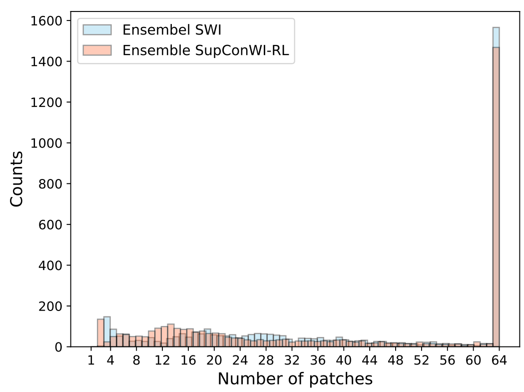


Figure 3. Comparison between Ensemble SWI [8] and the proposed **Ensemble SupConWI-RL** in the number of scanned patches

methods. As seen in Table 1, our SupConWI-RL outperforms SWI due to the additional supervised contrastive loss term both in terms of accuracy and average scanning time. The proposed SupCon-RL is also designed for the multi-defect case in contrast to SWI, which makes it a more general approach. It requires 38.0 measurements vs. 40.7 for SWI, which is equivalent to 4.2% of improvement if full scanning is considered as 100%. Also, accuracy and F1 score are comparable with pure ResNet-34 and ResNet-34 w/ SCL, which require all 64 patches.

In Fig. 3, we show the histogram for the number of measurements for both Ensemble SWI and our **Ensemble SupConWI-RL**. It is clear from the figure that our method reduces the number of full scan cases where all patches need to be scanned. Also, it shifts the number of scans closer to the left side of the x-axis. This means that our model gradually reduces the number of scans compared to SWI, without compromising the classification accuracy as Tab. 1 illustrates.

In the case of the multi-defect case, **Ensemble SupConWI-RL** requires 102.1 out of 169 patches for prediction, which means that it saves $\approx 40\%$ of scanning time. At the same time, accuracy and F1 score are comparable for the hybrid approach and pure CNN strategies. That means that **Ensemble SupConWI-RL** achieves the desired goal: to predict the defect type using as small a number of measurements as possible. We also compare it with ResNet-34 w/ SCL. However, we have not observed any improvement for the ResNet from the SCL. The purpose of this comparison is to show that our **Ensemble SupConWI-RL** can be used with any CNN-based approach in a hybrid scenario.

Our algorithm contains several integral parts, such as supervised contrastive loss, ensemble and hybridization with CNN. In Section 5, we consider the detailed contribution of every component to the outcome, while this subsection provides the best result for comparison.

4.3. Anomaly detection results

In practice, it can be very time-consuming and expensive to classify all defects during high-volume manufacturing. Therefore, one may need to find the first defect in which the model is certain, stop resource-consuming scanning at this moment, and label the wafer as an outlier. Thus, we can use the trained classification model by changing the stopping

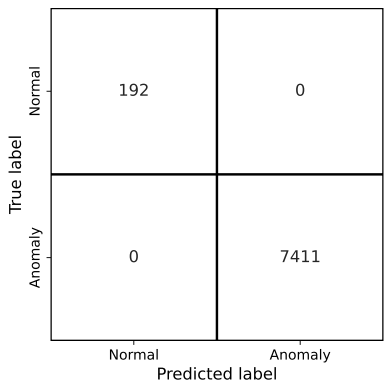


Figure 4. Confusion matrix for the anomaly prediction scenario

Table 3. Distribution over correct predictions

Correct normal	Correct anomaly	
	Correct type	Incorrect type
192	7401	10

criterion in Algorithm 1 without fine-tuning it to the outlier detection problem. It should be noted that, in contrast to the standard anomaly detection problem, the algorithm used both normal and defective classes. However, our goal in this subsection is to show additional advantages of the proposed feedback loop.

Fig. 4 shows the number of True Positive, True Negative, False Positive, and False Negative predictions with **Ensemble SupConWI-RL**. The figure shows that all wafers were correctly identified as normal and defective. The average and median numbers of scanning steps are 21.1 and 15 out of 169, corresponding to 12.5% and 9% respectively. That means if our goal is to identify wafers with defects, we can do it on average 8 times faster than CNN-based methods or any other ones requiring full scanning.

However, there are cases where the algorithm finds the wrong failure pattern on a defective surface. The algorithm may see a defect that is not present on the surface and therefore predict it as an outlier, but give the wrong reason for stopping. In Table 3, we demonstrate the numbers of correctly and wrongly identified failure patterns in the defective wafers. As a result, we have 10 wafers that were correctly identified as defective, but the decisive output defect is incorrect. Fortunately, this can be fixed further in the future, where all defects should be found. Overall, the model makes 10 mistakes out of 7603 examples, which is about 0.0013% of error.

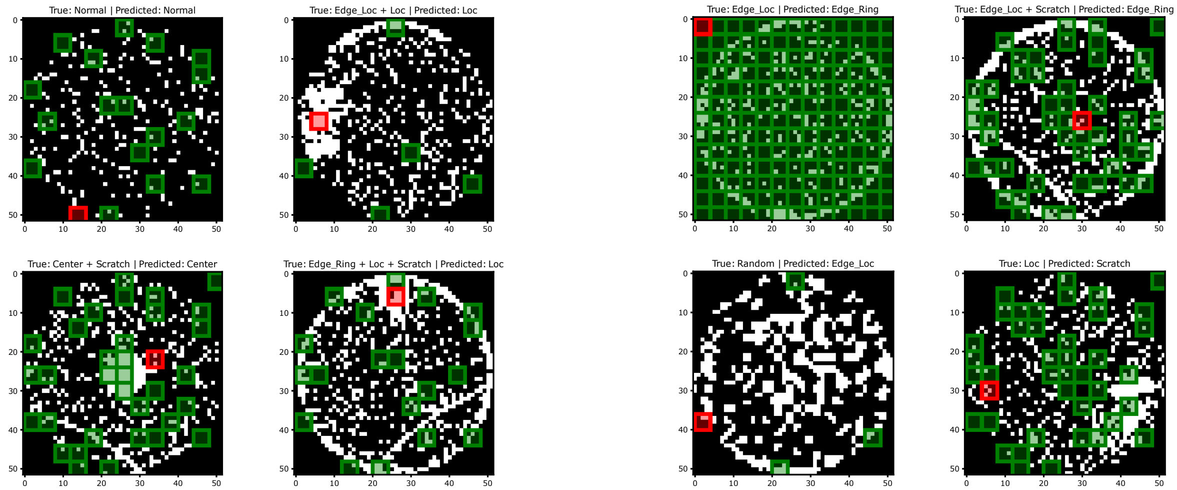
In Fig. 5, we visualize some examples where the algorithm is certain about (a) at least one correctly identified defect, and also examples where (b) the algorithm signals for a wrong defect in a defective surface. In the second case, we can see examples where the algorithm confuses *Edge Ring* and *Edge Loc* (see Fig. 5(b) up). However, we want to highlight that on these images for humans, it is also difficult to distinguish between *Edge Ring* and *Edge Loc*. Also, it sometimes struggles to find *Scratch* since it may allocate a small space on the surface or can be similar to a different defect pattern if it overlaps with something else.

5. Ablation study and discussion

In this section, we explore the importance of each component in the proposed approach. Thus, we compare the results of the proposed approach if we exclude one (or more) components from **Ensemble SupConWI-RL**. We consider the impact of the supervised contrastive loss (SCL), the ensemble and the combination with ResNet-34 on the final result. Table 4 summarizes our observations showing the average accuracy and stopping time step t^*. Each considered part significantly improves the algorithm's performance, bringing it closer to a pure CNN strategy. The table demonstrates the need for each component: SCL improves accuracy by 2% in the single model, even reducing the number of scanned patches. Also, the ensemble gains an additional 0.76% compared to the single-defect case, with almost no effect on the number of scanning steps. Finally, if the wafer requires full scanning, it is more beneficial to use CNN (e.g., ResNet-34) for prediction, which adds around 2%, resulting in 98.45% of accuracy.

Also, we would like to highlight the importance of an RL-based controller. Thus, we compare the result of **Single SupConWI-RL** and **Ensemble SupConWI-RL** with Single SupConWI and Ensemble SupConWI, where the patch order for every wafer surface is random. Table 5 shows the advantage of the RL-optimized patch sequence compared to random ones. As we can see from the table, the RL-optimized sequence saves us around 25 patches for scanning, which is about 15% out of the total number of patches. Moreover, in both single-model and ensemble cases, the accuracy is greater for the RL-optimized patch order, especially for the ensemble case, where the difference is 0.84%. Thus, patch order optimization has a positive impact not only on the number of measurements but also on the recognition capabilities of the model. We explain this by the fact that the model does not receive uninformative patches that could interfere with its decision.

One drawback of the ensemble strategy is the larger number of models compared to the single-model case. However, as it was also discussed in SWI [8], RNNs and MLPs that we use require significantly fewer training parameters than CNNs, so the memory costs in the hybrid approach are insignificant compared to those already allocated for CNN if M is not significantly large. In our case, $M = 5$, which results in 60% of additional memory costs compared



(a) (b)

Figure 5. Sequences of scanned patches until the stopping for (a) correctly and (b) incorrectly identified defects. The red colour shows the final patch.

Table 4. Ablation study on MixedWM-38 dataset: the usage of SCL, Ensemble and Hybrid approach improves accuracy by 5% without additional measurement cost t^*

Model	SCL	Ensemble	Hybrid	Acc (%)	t^*
Single WI-RL	✗	✗	✗	93.58	101.1/169
Single SupConWI-RL	✓	✗	✗	95.58	100.2/169
Ensemble WI-RL	✓	✓	✗	94.84	101.2/169
Ensemble SupConWI-RL	✓	✓	✗	96.34	102.1/169
ResNet-34 + Ensemble WI-RL	✗	✓	✓	97.44	101.2/169
ResNet-34 + **Ensemble SupConWI-RL**	✓	✓	✓	**98.45**	102.1/169

Table 5. Study on the need for an RL-optimized patch sequence: RL improves accuracy compared to random sequences

Model	RL	Acc (%)	t^*
Single SupConWI	✗	95.42	126.4/169
Single SupConWI-RL	✓	**95.58**	**100.2**/169
Ensemble SupConWI	✗	95.50	127.3/169
Ensemble SupConWI-RL	✓	**96.34**	**102.1**/169

to a pure ResNet-34 strategy.

6. Conclusion

In this work, we propose a method for the wafer surface defect inspection called SupConWI-RL. In contrast to previous work, our approach is suitable for scenarios with multiple defects on the wafer. Moreover, we incorporated the supervised contrastive loss into the training procedure, which also improved the recognition capabilities of the approach. Our SupConWI-RL requires, on average, 60.5% of the surface, while having comparable accuracy as CNN-based models, which need the full surface scan.

Additionally, our trained model can be used for anomaly detection without any adaptation. Once SupConWI-RL is trained for the multi-defect classification problem, one can use it to find defective surfaces that contain at least one defect. This way, the algorithm does not identify all failures present on the surface, but signals for defective wafers by detecting at least one defect present there. As a result, the model needs only 12.5% of the surface on average to identify anomalies with 100% accuracy.

We believe our work provides important findings for automatic visual inspection, demonstrating the effectiveness of the feedback loop for the problem of the multi-defect case as well. Our results show that a significant reduction in scanning time can be achieved without loss of accuracy.

Acknowledgement The project is supported by the Chips Joint Undertaking and its members, including the top-up funding by RVO (The Netherlands Enterprise Agency).

References

[1] Youngjae Bae and Seokho Kang. Supervised contrastive learning for wafer map pattern classification. *Engineering Applications of Artificial Intelligence*, 126:107154, 2023. 2, 5, 6

[2] Tim Bakker, Herke van Hoof, and Max Welling. Experimental design for mri by greedy policy search. *Advances in Neural Information Processing Systems*, 33:18954–18966, 2020. 2, 3

[3] Shouhong Chen, Yuxuan Zhang, Xingna Hou, Yuling Shang, and Ping Yang. Wafer map failure pattern recognition based on deep convolutional neural network. *Expert Systems with Applications*, 209:118254, 2022. 2

[4] Shouhong Chen, Zhentao Huang, Tao Wang, Xingna Hou, and Jun Ma. Mixed-type wafer defect detection based on multi-branch feature enhanced residual module. *Expert Systems with Applications*, 242:122795, 2024. 2

[5] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2014. 3

[6] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. *Advances in Neural Information Processing Systems*, 32, 2019. 3, 4

[7] Aleksandr Dekhovich and Miguel A Bessa. Continual learning for surface defect segmentation by subnetwork creation and selection. *Journal of Intelligent Manufacturing*, pages 1–15, 2024. 1

[8] Aleksandr Dekhovich, Oleg Soloviev, and Michel Verhaegen. Sequential wafer map inspection via feedback loop with reinforcement learning. *Expert Systems with Applications*, page 126996, 2025. 1, 2, 3, 4, 5, 6, 7

[9] Lie Dou and M-P Broderick. A new technique for automated wafer inspection and classification of particles and crystalline defects. In *1997 IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop ASMC 97 Proceedings*, pages 180–184. IEEE, 1997. 1

[10] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 3

[11] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 3

[12] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30, 2017. 3

[13] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990. 4

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 6

[15] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016. 3

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3

[17] Jianchuan Huang, Kuo-Yi Lin, Jia Xu, et al. Imbalanced wafer map dataset classification with semi-supervised learning method and optimized loss function. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 1815–1819. IEEE, 2022. 2

[18] YongSung Ji and Jee-Hyong Lee. Using gan to improve cnn performance of wafer map defect type classification: Yield enhancement. In *2020 31st annual SEMI advanced semiconductor manufacturing conference (ASMC)*, pages 1–6. IEEE, 2020. 2

[19] Seokho Kang. Rotation-invariant wafer map pattern classification with convolutional neural networks. *IEEE Access*, 8: 170650–170658, 2020. 2

[20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 2

[21] Minkyu Kim, QHwan Kim, Kyu-Baik Chang, Jaehoon Jeong, Sunghee Lee, Seonghui Mo, Dahan Kang, Jinkook Park, Young-Seok Kim, Yongdeok Jeong, et al. A few-shot machine learning-based ocd metrology algorithm with anomaly detection and wafer-level data augmentation. In *Metrology, Inspection, and Process Control XXXVIII*, pages 189–196. SPIE, 2024. 1

[22] Kiryong Kyeong and Heeyoung Kim. Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks. *IEEE Transactions on Semiconductor Manufacturing*, 31(3):395–402, 2018. 2

[23] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 3

[24] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *ArXiv e-prints*, pages arXiv–1607, 2016. 5

[25] Jialin Li, Ran Tao, Renxiang Chen, Yongpeng Chen, Chengying Zhao, and Xianzhen Huang. Sample-imbalanced wafer map defects classification based on auxiliary classifier denoising diffusion probability model. *Computers & Industrial Engineering*, 192:110209, 2024. 2

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. 2

[27] Takeshi Nakazawa and Deepak V Kulkarni. Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing*, 31(2):309–314, 2018. 2

[28] Mehrdad Nikoonahad, Charles E Wayman, and Stephen A Biellak. Defect detection algorithm for wafer inspection

based on laser scanning. *IEEE transactions on semiconductor manufacturing*, 10(4):459–468, 1997. 1

[29] Jacques Noom, Nguyen Hieu Thao, Oleg Soloviev, and Michel Verhaegen. Closed-loop active model diagnosis using bhattacharyya coefficient: Application to automated visual inspection. In *International Conference on Intelligent Systems Design and Applications*, pages 657–667. Springer, 2020. 1

[30] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2):1–38, 2021. 1

[31] Luis Pineda, Sumana Basu, Adriana Romero, Roberto Calandra, and Michal Drozdzal. Active mr k-space sampling with reinforcement learning. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*, pages 23–33. Springer, 2020. 2

[32] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2, 4

[33] NG Shankar and ZW Zhong. Defect detection on semiconductor wafer surfaces. *Microelectronic engineering*, 77(3-4): 337–346, 2005. 1

[34] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 2

[35] Xian Tao, Xinyi Gong, Xin Zhang, Shaohua Yan, and Chandranath Adak. Deep learning for unsupervised anomaly localization in industrial images: A survey. *IEEE Transactions on Instrumentation and Measurement*, 71:1–21, 2022. 1

[36] Tsung-Han Tsai and Yu-Chen Lee. A light-weight neural network for wafer map classification based on data augmentation. *IEEE Transactions on Semiconductor Manufacturing*, 33(4):663–672, 2020. 2

[37] Javier Villalba-Diez, Daniel Schmidt, Roman Gevers, Joaquín Ordieres-Meré, Martin Buchwitz, and Wanja Wellbrock. Deep learning for industrial computer vision quality control in the printing industry 4.0. *Sensors*, 19(18):3987, 2019. 1

[38] Junliang Wang, Chuqiao Xu, Zhengliang Yang, Jie Zhang, and Xiaoou Li. Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition. *IEEE Transactions on Semiconductor Manufacturing*, 33(4):587–596, 2020. 2

[39] Shuyu Wang, Zhitao Zhong, Yuliang Zhao, and Lei Zuo. A variational autoencoder enhanced deep learning model for wafer defect imbalanced classification. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 11(12):2055–2060, 2021. 2

[40] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992. 2

[41] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020. 3

[42] Ming-Ju Wu, Jyh-Shing R Jang, and Jui-Long Chen. Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Transactions on Semiconductor Manufacturing*, 28(1):1–12, 2014. 2

[43] Chen-Yu Yen, Raghav Singhal, Umang Sharma, Rajesh Ranganath, Sumit Chopra, and Lerrel Pinto. Adaptive sampling of k-space in magnetic resonance for rapid pathology prediction. *arXiv preprint arXiv:2406.04318*, 2024. 1

[44] Tianwei Yin, Zihui Wu, He Sun, Adrian V Dalca, Yisong Yue, and Katherine L Bouman. End-to-end sequential sampling and reconstruction for mri. *arXiv preprint arXiv:2105.06460*, 2021. 2

[45] Naigong Yu, Qiao Xu, and Honglu Wang. Wafer defect pattern recognition and analysis based on convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing*, 32(4):566–573, 2019. 2