arXiv:2610.40421v1 [cs.LG] 29 Oct 2026

Uncertainty-aware Feedback Loop Diagnosis: integrating reinforcement learning and conformal prediction for reliable wafer defect classification

Aleksandr Dekhovich*
TU Delft
a.dekhovich@elsevier.com

Carlas Smith
TU Delft

Oleg Soloviev
TU Delft & Flexible Optical B.V.

Abstract

Semiconductor manufacturing is a complex, multi-step process that requires both high precision and low operating costs. Therefore, detecting defects in a high-volume setting is an important component of wafer map analysis for reliable production. However, most state-of-the-art deep learning approaches require a full scan of the surface, which hinders the development of high-throughput inspection systems. To address this problem, we propose Uncertainty-aware Feedback Loop Diagnosis (UFLD), a novel algorithm that employs reinforcement learning for adaptive measurement acquisition and provides a prediction uncertainty level after each step, stopping the process once sufficient information is obtained. Unlike existing methods that rely on ensemble models or auxiliary confidence estimation deep learning models, UFLD utilizes a more parameter-efficient conformal prediction framework to determine the optimal stopping time. The proposed UFLD reduces acquisition time by about 40% and achieves 98.4% inspection accuracy, without requiring an additional learned confidence-estimation network. We validated our approach on the MixedWM38 dataset, demonstrating its advantages over existing adaptive acquisition procedures and standard convolutional neural networks.

1 Introduction

The semiconductor industry's evolution is historically driven by Moore's Law, the empirical observation that the number of transistors in an integrated circuit doubles approximately every two years [1]. Due to the increasing demand for chips, defect detection plays a crucial role in semiconductor manufacturing, ensuring the quality of chips produced [2]. However, this task is even more challenging in real-time high-volume manufacturing [3], which led to the development of automated defect inspection systems [4, 5]. In the past decade, artificial intelligence (AI), and specifically deep learning (DL), algorithms have achieved remarkable progress in a range of tasks, including industrial applications [3, 6, 7]. A significant portion of these works aims to develop vision-based defect detection systems to analyze the surface of interest by classifying [8, 9], segmenting [10, 11], or detecting [12, 13] possible defective regions.

An efficient inspection system should be accurate, reliable and fast at the same time. Therefore, despite the success of convolutional neural networks (CNNs) in classification problems, CNN-based solutions typically require a full scan of the wafer. In modern high-volume manufacturing, collecting high-resolution data for every die is expensive and time-consuming. Consequently, there is a need for alternative approaches employing adaptive data acquisition (ADA) strategies that can maintain high classification accuracy while minimizing the number of scanned regions.

*Corresponding author. This work was done while the author was a postdoctoral researcher at TU Delft.

Preprint.

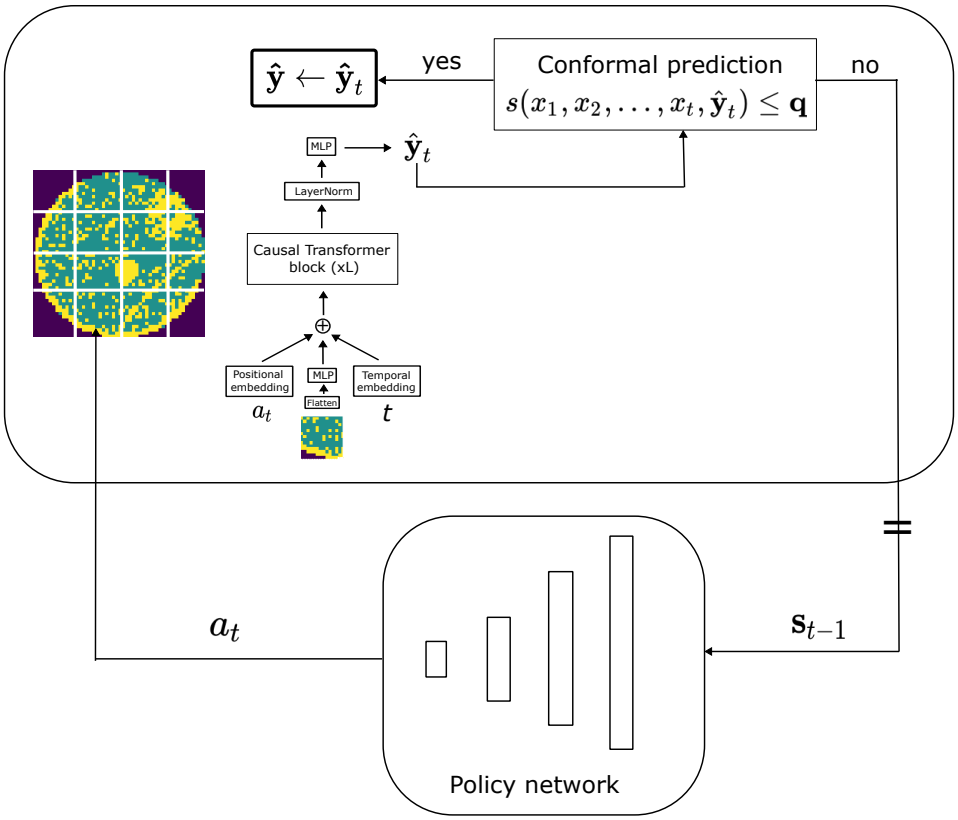


Figure 1: An overview of UFLD: the Causal Transformer predicts the defect type from the available patches; conformal prediction evaluates whether the predicted label is in the prediction set. If yes, the scanning process stops; otherwise, the policy network selects the next most informative patch, continuing the feedback loop.

To address this, we propose Uncertainty-aware Feedback Loop Diagnosis (UFLD) that uses a reinforcement learning (RL) strategy for ADA and a conformal prediction-based framework [14] to evaluate the trustworthiness of the prediction after every iteration of the acquisition process to mitigate the limitations of the previous works [15, 16]. Figure 1 illustrates the algorithm: the Causal Transformer plays the role of the classifier to predict the defect pattern from the available observations, while the RL-based policy network adaptively selects the next region for scanning. After each step, a simple threshold-based rule determines whether the acquisition process can be stopped. The main contributions of our work can be summarized as follows:

- We developed a feedback loop algorithm to increase the throughput of inspection tools, while maintaining the performance level of CNN models that require full scanning. The code implementation of the work is publicly available on GitHub: https://github.com/adekhovich/UFLD.
- In contrast to previous works on sequential wafer inspection where RNN-based classifiers were used, we utilize a causal transformer-based model that has better predictive capabilities. A single transformer model achieves performance comparable to an ensemble of RNNs, while requiring fewer training parameters.
- We introduce a novel stopping rule for the adaptive data acquisition process based on the conformal prediction framework. Our approach reduces algorithmic complexity compared to previous works by replacing an auxiliary confidence-estimator network with a simple threshold-based rule without sacrificing prediction quality.

The rest of the article is organized as follows: in Section 2, we provide the necessary background for the work, Section 3 describes the proposed feedback loop algorithm, while Section 4 shows the results and compares with other methods, and Section 5 concludes the article.

2 Related work

Wafer defect recognition using deep learning Deep learning (DL) models are widely used in wafer failure pattern detection [17, 18, 19]. Convolutional neural networks (CNNs) have been trained under different scenarios, namely zero- and few-shot learning [20, 21] to meet the real-world data scarcity constraints, as well as transfer learning [22, 23] and supervised contrastive learning [24] for better generalization. In addition, some works consider unsupervised learning tasks for wafer map analysis with CNNs [25]. For instance, the active cluster framework [26] has been proposed for annotating unlabeled wafer maps. Furthermore, Nakazawa & Kulkarni [17] developed a CNN-based encoder–decoder model for abnormal defect pattern detection and segmentation of previously unseen real wafer maps.

Transformer-based architectures have been successfully used in defect classification tasks [27, 28]. By leveraging self-attention layers, ViTs [29] often generalize better than CNNs; however, they may require more training data for convergence [30]. Therefore, fine-tuning of the pretrained ViT is a better strategy than training a model from scratch [31, 32]. Recently, Liao *et al*. proposed WaferDSR-RAG [33], which uses retrieval-augmented generation (RAG) [34]. In WaferDSR-RAG, this technique is used for an LLM with a specialized semiconductor knowledge base, linking visual defect patterns to historical engineering logs.

In the discussed works, both CNNs and Transformer-based architectures require a full wafer map and do not process partial observations from the surface, which makes them impractical for real-time high-volume manufacturing. Consequently, we further discuss approaches that predict a label by adaptively selecting the most informative regions, which significantly reduces overall scanning time.

Adaptive data acquisition Adaptive data acquisition is a widely used technique to reduce the cost of measurement collection in high-cost processes. In scanning microscopy, Ye *et al.* [35] introduced a closed-loop denoising system with pixel-wise uncertainty estimation based on conformal prediction, where only the most uncertain regions of an image are adaptively selected for rescanning. Holman *et al.* [36] developed a grid-less autonomous adaptive sampling method for scanning hyperspectral imaging. By utilizing hybrid sequential design strategies for surrogate model construction, their system identifies regions of steeper physico-chemical gradients and increases sampling density in those areas in real-time. This approach significantly reduces acquisition time compared to traditional uniform grid sampling while maintaining high fidelity in complex biological samples.

Several works employ reinforcement learning (RL) [37] for adaptive data acquisition [38, 39, 15], modeling the problem as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \mu, \gamma)$. In this framework, $\mathcal{S}$ represents the state space, $\mathcal{A}$ the action space, $P(s_{t+1}|s_t, a_t)$ the transition distribution, $R(s_t, a_t)$ the reward function, $\mu(s_0)$ the initial state distribution, and $\gamma \in (0, 1]$ the discount factor. The agent's objective is to learn an optimal policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that maximizes the expected cumulative return, given by $\mathbb{E}_\pi \left[\sum_t \gamma^t R(s_t, a_t)\right]$, where actions and next states are sampled according to $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$, respectively.

For instance, Bakker *et al*. [40] utilize a policy gradient RL method for adaptive sampling in k-space for MRI reconstruction, while Yen *et al*. [39] developed Adaptive Sampling for MR (ASMR) for image-based disease detection. However, these methods predefine the number of measurements that are adaptively selected. The recently proposed SupConWI-RL [16] adaptively selects the next patch and stops the process when an auxiliary confidence estimator model signals that the collected information is sufficient for a reliable prediction. As a result, the number of scanned patches is not predefined and depends on how difficult the failure pattern is to classify.

Conformal prediction Conformal prediction (CP) is a data-based, distribution-free technique for constructing prediction sets, using a calibration set $\mathcal{D}_{\text{calib}} = \{(x_i, y_i)\}_{i=1}^{n_{\text{calib}}}$ that is different from the train or test set [41]. The key assumption for the coverage guarantee is the exchangeability property. In addition, it is worth noting that the i.i.d. assumption is often used since exchangeability follows from i.i.d., while the opposite is not true. Vanilla CP consists of the following steps:

1. select a non-conformity scoring function $s : X \times Y \to [0, 1]$;
2. calculate non-conformity scores $s_1, s_2, \ldots, s_{n_{\text{calib}}}$, where $s_i = s(x_i, y_i),\ (x_i, y_i) \in \mathcal{D}_{\text{calib}}$;

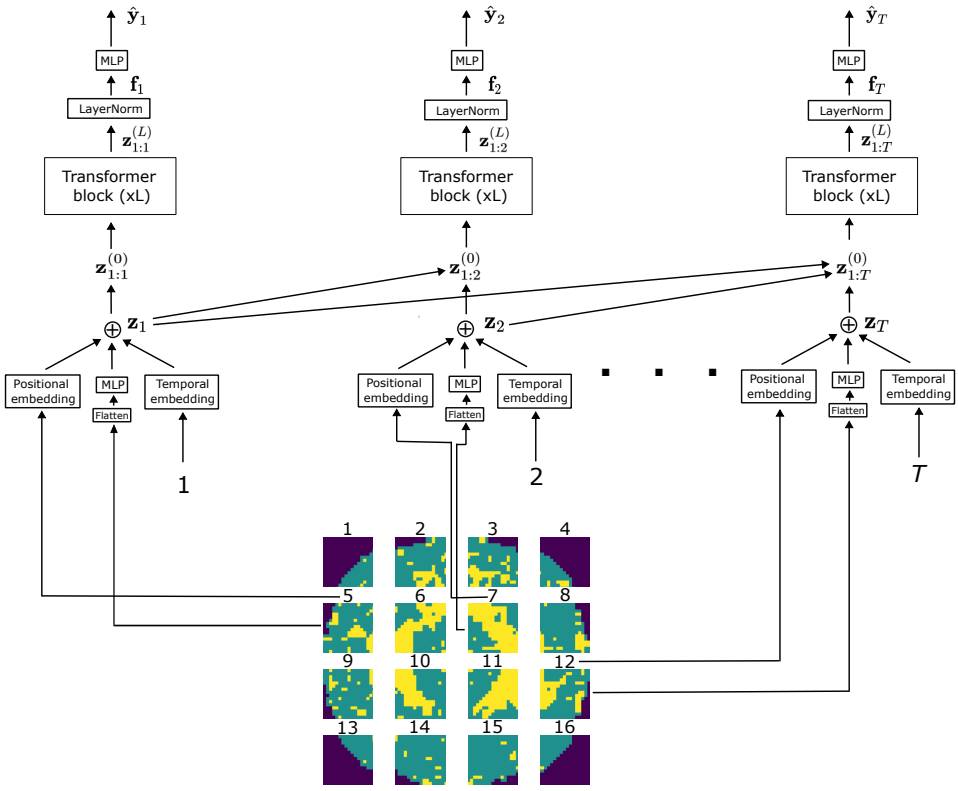


Figure 2: An illustration of the transformer-based classifier: at every time step t it aggregates t embedded patches along with their positional and temporal embedding, followed by L Transformer blocks. Finally, the layer normalization and MLP are applied to obtain the output.

3. calculate the empirical quantile $\hat{q}_\alpha = \frac{\lceil (n_{\text{calib}}+1)(1-\alpha) \rceil}{n_{\text{calib}}}$ of the scores $s_1, s_2, \ldots, s_{n_{\text{calib}}}$, where $1-\alpha$ is the user-defined coverage level;

4. construct the prediction set for a test point x_{test} with a miscoverage level α: $C_\alpha(x_{\text{test}}) = \{y : s(x_{\text{test}}, y) \leq \hat{q}_\alpha\}$.

Vovk *et al.* [14] provide theoretical guarantees for the prediction sets depending on the size of the calibration set n_{calib}: $\mathbb{P}\big(y \in C_\alpha(x_{\text{test}})\big) \geq 1-\alpha$, for the true label y of the test point x_{test}. Adaptive Prediction Sets (APS) [42, 43] and Conformal Risk Control (CRC) [44] have been proposed as extensions of the vanilla CP. Like vanilla CP, APS guarantees marginal coverage, while its adaptive conformity score allows the prediction-set size to vary with the difficulty of each input. Furthermore, CRC generalizes vanilla CP to the scenario where the output is high-dimensional by controlling an average loss, e.g., an image segmentation task with a binary mask.

3 Proposed method: Uncertainty-aware feedback loop diagnosis

Our Uncertainty-aware Feedback Loop Diagnosis (UFLD) framework consists of three core components: a Transformer-based classifier, an RL-based adaptive patch selector, and a conformal prediction-based stopping rule. The goal of UFLD is to optimize the input sequence to minimize the number of measurements. To simulate real-world conditions, we split an input image, a wafer map, into patches that can be treated as dies. Formally, we denote the input as $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, where $\mathbf{x}_i$ is a patch. In our algorithm, the RL-based selector iteratively selects the next patch based on prior observations, while the Transformer-based classifier predicts the defect type after each acquisition. The loop continues until the conformal prediction-based stopping rule determines that the classification reaches a reliability threshold.

Transformer-based classifier First, we pretrain our classifier on randomly sampled sequences of patches $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_T}$, so the Transformer model can provide accurate predictions across different acquisition trajectories that the RL agent explores during optimization. To achieve this sequence robustness, a standard causal Transformer is augmented with a learnable sequence-step embedding $\mathbf{E}_t$. This embedding encodes the temporal position of a patch within the acquisition sequence, supplementing spatial positional embeddings $\mathbf{E}_{i_t}$ of the patch on the wafer map. Figure 2 represents the architecture of the classifier. Overall, the formal computations within the model to compute output $\hat{\mathbf{y}}_t$ for $1 \leq t \leq T$ are defined as follows:

$$\begin{aligned}
\mathbf{z}_s &= \texttt{MLP}(\texttt{Flatten}(\mathbf{x}_{i_s})) + \mathbf{E}_{i_s} + \mathbf{E}_s, \quad 1 \leq s \leq t \\
\mathbf{z}_{1:t}^{(0)} &:= [\mathbf{z}_1\ \mathbf{z}_2 \ldots \mathbf{z}_t] \\
\mathbf{z}_{1:t}^{(l)} &= \texttt{TransformerBlock}\left(\mathbf{z}_{1:t}^{(l-1)}\right), \quad l = 1, 2, \ldots, L, \\
\mathbf{f}_t &= \texttt{LayerNorm}\left(\mathbf{z}_{1:t}^{(L)}\right) \\
\hat{\mathbf{o}}_t &= \texttt{MLP}\big(\mathbf{f}_t\big).
\end{aligned} \tag{1}$$

Since each wafer map can contain up to four defects in the MixedWM38 dataset, we are solving a multi-label classification problem. To address this, we apply multi-hot encoding to the labels, resulting in a target vector $\mathbf{y} \in \{0,1\}^C$, where C denotes the total number of possible defect classes. Therefore, the output of the classifier is a probability tensor $\mathbf{p}_t$ that is calculated by applying the $\texttt{Sigmoid}$ function to each component of the logit tensor $\hat{\mathbf{o}}_t$: $\mathbf{p}_t = \texttt{Sigmoid}(\hat{\mathbf{o}}_t)$.

Then, we obtain the prediction $\hat{\mathbf{y}}_t$ by thresholding the probability tensor $\mathbf{p}_t = (p_{t,1}, p_{t,2}, \ldots, p_{t,C})$: $\hat{\mathbf{y}}_t = \big(\mathbf{1}\{p_{t,1} \geq 0.5\}, \mathbf{1}\{p_{t,2} \geq 0.5\}, \ldots, \mathbf{1}\{p_{t,C} \geq 0.5\}\big)$.

To optimize the parameters of the Transformer, we use standard binary cross-entropy loss for each type of defect. Several works have shown the effectiveness of the supervised contrastive learning loss term for the classification task [45], including sequential wafer map defect recognition [24, 11]. Therefore, we include it in the total loss that we optimize:

$$\mathcal{L}(\theta_{\text{classifier}}) = \frac{1}{|\mathcal{B}_{\text{aug}}|}\frac{1}{T}\sum_{(\mathbf{X},\mathbf{y})\in\mathcal{B}_{\text{aug}}}\sum_{t=1}^{T}\Bigg(\text{BCELoss}(\theta_{\text{classifier}}; \mathbf{p}_t, \mathbf{y}) + \lambda \cdot \text{SCLoss}(\theta_{\text{classifier}}; \{\mathbf{f}_t\}_{i=1}^{|\mathcal{B}|}, \{\tilde{\mathbf{f}}_t\}_{i=1}^{|\mathcal{B}|}, \mathbf{y})\Bigg),$$

where batch $\mathcal{B}_{\text{aug}} = \mathcal{T}_1(\mathcal{B}) \cup \mathcal{T}_2(\mathcal{B})$ is obtained by applying two data augmentations $\mathcal{T}_1$ and $\mathcal{T}_2$ to the original training batch $\mathcal{B}$, $\{\mathbf{f}_t\}_{i=1}^{|\mathcal{B}|}$ and $\{\tilde{\mathbf{f}}_t\}_{i=1}^{|\mathcal{B}|}$ are the last feature representations of sequences up to time t of $\mathcal{T}_1(\mathcal{B})$ and $\mathcal{T}_2(\mathcal{B})$ respectively.

RL-based patch selection Once the classifier is trained, our goal is to optimize the acquisition sequence to minimize the number of measurements (scanned patches) required for each wafer map. Therefore, we train a policy network $\pi_{\theta_{\text{RL}}}$ that selects the next patch from the set of available candidates. During training, the next action (patch) a_t is sampled from a categorical distribution with parameters $\pi_{t+1} = \pi_{\theta_{\text{RL}}}(\mathbf{s}_t) \odot \mathbf{m}_t$: $a_{t+1} \sim \text{Categorical}(\pi_{t+1})$, where $\mathbf{m}_t \in \{0,1\}^T$ is a binary mask representing the remaining unselected patches and $\mathbf{s}_t = (\mathbf{f}_t, \mathbf{m}_t)$. At initial step $t = 0$, $\mathbf{m}_0 = (1, 1, \ldots, 1)$ and $\mathbf{f}_0 = (0, 0, \ldots, 0)$. During inference, the agent follows a deterministic policy by selecting the available action with the maximum probability: $a_t = \text{argmax}\ \pi_t$. The reward function is designed as follows: $r_t = \begin{cases} 1 - c_{\text{RL}}, & \text{if } \hat{\mathbf{y}}_t = \mathbf{y} \\ -c_{\text{RL}}, & \text{otherwise,} \end{cases}$ where c_{RL} is a penalty term for every step. We should note that our reward function is different from the one in [16] because ours only depends on the correctness of the prediction, and not the score of some auxiliary confidence estimator, which may be unreliable or poorly calibrated.

Conformal prediction as stopping rule The main challenge in the proposed acquisition process is determining the stopping time t where the classification model is sufficiently certain to use the

prediction $\hat{\mathbf{y}}_t$. Stopping too early may result in inaccurate predictions, while delaying the stopping point will increase the data collection costs. The conformal prediction framework provides the opportunity to control the coverage rate $1-\alpha$. However, the application of conformal prediction to sequential decision-making is complicated by the potential violation of the exchangeability property, which is the fundamental assumption for the calibration and test datasets. In our case, the predictions $\hat{\mathbf{y}}_t$ are computed from partial observations $\mathbf{x}_{1:t}$. We ensure that the calibration set $\mathcal{D}_{\text{calib}}$ is processed in the same manner as the test set $\mathcal{D}_{\text{test}}$. Therefore, we should calculate the non-conformity scores and, as a result, the threshold $\hat{q}_\alpha$ on the calibration set using trajectories generated by the same policy across the same time steps as the test set.

In our work, we focus on point accuracy, where each component of the predicted label is correct. Therefore, our objective is the per-class certainty in the classifier output $\hat{\mathbf{y}}_{t,i},\ i = 1, 2, \ldots, C$ at time step t. For a given calibration point $(\mathbf{x}^i, \mathbf{y}^i) \in \mathcal{D}_{\text{calib}}$, we define non-conformity scores at time step t as follows:

$$S^i_{t,j}(\mathbf{x}^i, \mathbf{y}^i) = \begin{cases} 1 - p^i_{t,j}, & \text{if } y^i_j = 1 \\ p^i_{t,j}, & \text{if } y^i_j = 0, \end{cases} \tag{2}$$

where $p^i_{t,j}$ is the j-th component of the classifier's output probability distribution $\mathbf{p}_t$, and $\mathbf{y}^i = (y^i_1, y^i_2, \ldots, y^i_C)$.

Our approach relies on two sets of label-wise quantiles $\hat{\mathbf{q}}_y = (\hat{q}_{y,1}, \hat{q}_{y,2}, \ldots, \hat{q}_{y,C}),\ y \in \{0, 1\}$ for positive and negative classes. These quantiles play the role of thresholds, and we further call them "negative" and "positive" thresholds. These thresholds are computed through the calibration procedure. During inference, for a new test input $\mathbf{x}$, we calculate non-conformity scores

$$\begin{aligned} S_{0,t,j}(\mathbf{x}_{1:t}) &= p_{t,j}, \\ S_{1,t,j}(\mathbf{x}_{1:t}) &= 1 - p_{t,j} \end{aligned} \tag{3}$$

after every acquisition step t. We simultaneously track these values for each label $j = 1, 2, \ldots, C$, and our goal is to stop when the prediction set for each label has cardinality 1. Let us denote $E_{t,j} := \{S_{0,t,j} \le \hat{q}_{0,j} \wedge S_{1,t,j} > \hat{q}_{1,j}\} \vee \{S_{0,t,j} > \hat{q}_{0,j} \wedge S_{1,t,j} \le \hat{q}_{1,j}\}$ for each label $j = 1, 2, \ldots, C$. Therefore, our requirement of the size of the prediction set for label j is equivalent to $E_{t,j} = 1$. We need to stop the acquisition process when we are sure about all the predicted labels, which means that $E_t := E_{t,1} \wedge E_{t,2} \wedge \ldots \wedge E_{t,C} = 1$, and, therefore, every event $E_{t,j}$ is true. This means that the non-conformity score of every label is lower than the corresponding component of precisely one of the thresholds $\hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1$. As a result, we can formulate our RL acquisition process with the conformal prediction stopping rule in Algorithm 1.

Due to the sequential nature of our problem and differences in stopping times for each of the wafers, the quantiles $\hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1$ have to be computed from the non-conformity scores obtained from different time steps to better align the calibration setting with inference. However, at first, we do not have a rule to stop the acquisition process and collect the non-conformity scores for calibration. Thus, our calibration process consists of two parts: first, we compute the preliminary quantiles $\hat{\mathbf{q}}^A_0, \hat{\mathbf{q}}^A_1$ using the dataset $\mathcal{D}_A \subset \mathcal{D}_{\text{calib}}$ and by collecting the non-conformity scores $\mathbf{S}^i_T$ at the final step T. Second, we recalibrate these quantiles using them as a stopping criterion for the learned acquisition policy π on dataset $\mathcal{D}_B = \mathcal{D}_{\text{calib}} \setminus \mathcal{D}_A$. For each wafer in $\mathcal{D}_B$, non-conformity scores are then collected at the induced stopping time and used to compute the refined thresholds $\hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1$. Algorithm 4 combines these stages into the full calibration algorithm. In this way, the calibration process reproduces the inference setting, where predictions may be produced at input-dependent stopping times. Algorithm 2 illustrates the first stage of the calibration process, where we collect the non-conformity scores at time T. In Algorithm 3, we use $\hat{\mathbf{q}}^A_0$ and $\hat{\mathbf{q}}^A_1$ to construct a stopping rule and to collect the non-conformity scores at the end of the scanning process, which can be at an arbitrary time step.

The fact that different calibration samples stop at different acquisition times does not break exchangeability. Conditional on the preliminary thresholds and the acquisition policy, the stopping time is a deterministic function of the observed wafer trajectory. Hence, each calibration score is obtained by applying the same stopping time scoring rule to an exchangeable sample. The resulting stopping time scores are therefore exchangeable, even though their realized stopping times may differ. We note, however, that the resulting coverage guarantee applies directly to predictions evaluated under the stopping rule induced by the preliminary thresholds. When the refined thresholds are subsequently used to modify the stopping rule at inference, the standard split conformal exchangeability argument no longer provides an exact finite-sample coverage guarantee for the resulting adaptive procedure.

Algorithm 1 Inference with adaptive data acquisition and conformal prediction

function PREDICT(transformer classifier $\mathcal{F}$, acquisition policy π, test point $\mathbf{x}$, quantiles $\hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1$)
  $\hat{\mathbf{y}} \leftarrow \emptyset$
  $\mathbf{m}_0 \leftarrow (1, 1, \ldots, 1)$
  $\mathbf{f}_0 \leftarrow (0, 0, \ldots, 0)$
  for $t = 0, 1, \ldots, T-1$ **do**
    Initialize current state: $\mathbf{s}_t \leftarrow [\mathbf{f}_t, \mathbf{m}_t]$
    Predict next patch location: $a_{t+1} \leftarrow \pi(\mathbf{s}_t)$
    $\mathbf{m}_{t+1} \leftarrow \mathbf{m}_t$
    $\mathbf{m}_{t+1}[a_{t+1}] \leftarrow 0$
    $\mathbf{x}_{1:t+1} \leftarrow [x_{a_1}, x_{a_2}, \ldots, x_{a_{t+1}}]$
    $\hat{\mathbf{y}}_{t+1}, \mathbf{f}_{t+1} \leftarrow \mathcal{F}(\mathbf{x}_{1:t+1})$
    Compute $\mathbf{S}_{0,t+1}(\mathbf{x}_{1:t+1})$ and $\mathbf{S}_{1,t+1}(\mathbf{x}_{1:t+1})$ using (3)
    if $E(\mathbf{S}_{0,t+1}, \mathbf{S}_{1,t+1}, \hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1) = 1$ **then**
      $\hat{\mathbf{y}} \leftarrow \hat{\mathbf{y}}_{t+1}$
      break loop
    end if
  end for
  if $\hat{\mathbf{y}} = \emptyset$ **then**
    $\hat{\mathbf{y}} \leftarrow \mathcal{F}(\mathbf{x}_{1:T})$
  end if
  return $\hat{\mathbf{y}}$
end function

Algorithm 2 Calibrate at time step T

function CALIBRATE_A(transformer classifier $\mathcal{F}$, calibration dataset $\mathcal{D}_A = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^{n_A}$, miscoverage rate α)
  Prediction at final step T: $\hat{\mathbf{y}}^i = \mathcal{F}(\mathbf{x}^i) \quad i = 1, 2, \ldots, n_A$
  Compute non-conformity scores $\mathbf{S}_y^i = (S_{y,1}^i, S_{y,2}^i, \ldots, S_{y,C}^i)$, $y = 0, 1$, using Equation (2)
  Compute per-class quantiles $\hat{\mathbf{q}}_y^A = (\hat{q}_{y,1}^A, \hat{q}_{y,2}^A, \ldots, \hat{q}_{y,C}^A)$, where:

$$\hat{q}_{y,j}^A = \text{Quantile}\left(\{S_{y,j}^i : y_j^i = y\}_{i=1}^{n_{\mathrm{A}_{y,j}}}; \frac{\lceil (n_{\mathrm{A}_{y,j}}+1)(1-\alpha) \rceil}{n_{\mathrm{A}_{y,j}}}\right), \quad n_{\mathrm{A}_{y,j}} = \left|\{i : y_j^i = y\}\right|$$

  return $\hat{\mathbf{q}}_0^A, \hat{\mathbf{q}}_1^A$
end function

Remark 3.1 *In our setting, the conformal thresholds $\hat{\mathbf{q}}_0$ and $\hat{\mathbf{q}}_1$ are used as certification thresholds. Therefore, from the definition of non-conformity scores in Equation (2) an increase of α leads to a decrease of $\hat{q}_{y,j},\ y \in \{0, 1\},\ j = 1, 2, \ldots, C$. As a result, it is harder to certify labels, and more labels remain undecided. Hence, the model requires more patches for scanning. This means that smaller thresholds lead to fewer early certifications, and therefore stopping happens later.*

Remark 3.2 *In some cases, the condition provided by the conformal prediction framework does not hold for any $t \in \{1, 2, \ldots, T\}$. This results in the acquisition process continuing until all measurements (patches) have been obtained. If this occurs for a given wafer map, we can utilize a CNN model to predict the defect type. Since all patches are already available, this does not increase the number of scanning steps but can significantly improve inspection performance. A system that works in this manner is referred to as a hybrid model (a similar approach has been used in [16]).*

4 Experiments

4.1 Experimental setup

Dataset We validated our method on the MixedWM38 dataset, which contains around $38{,}000$ images of wafer maps with one, two, three and four defects, as well as normal ones. Each input image in the MixedWM38 dataset has a size of 52×52, which we divide into $T = 169$ patches of size 4×4. Before that, we apply two sets of augmentation transformations, $\mathcal{T}_1$ and $\mathcal{T}_2$, which are required

Algorithm 3 Recalibrate using policy π

function CALIBRATE_B(transformer classifier $\mathcal{F}$, acquisition policy π, calibration dataset $\mathcal{D}_B = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^{n_B}$, preliminary quantiles $\hat{\mathbf{q}}_0^A, \hat{\mathbf{q}}_1^A$, miscoverage rate α)
  /* Set of "negative" non-conformity scores */
  $S_0[j] \leftarrow \emptyset,\ j = 1, 2, \ldots, C.$
  /* Set of "positive" non-conformity scores */
  $S_1[j] \leftarrow \emptyset,\ j = 1, 2, \ldots, C.$
  for $i = 1, 2, \ldots, n_B$ **do**
    $\mathbf{m}_0 \leftarrow (1, 1, \ldots, 1)$
    $\mathbf{f}_0 \leftarrow (0, 0, \ldots, 0)$
    for $t = 0, 1, \ldots, T-1$ **do**
      Initialize current state: $\mathbf{s}_t \leftarrow [\mathbf{f}_t, \mathbf{m}_t]$
      Predict next patch location: $a_{t+1} \leftarrow \pi(\mathbf{s}_t)$
      $\mathbf{m}_{t+1} \leftarrow \mathbf{m}_t$
      $\mathbf{m}_{t+1}[a_{t+1}] \leftarrow 0$
      $\mathbf{x}_{1:t+1}^i \leftarrow [x_{a_1}^i, x_{a_2}^i, \ldots, x_{a_{t+1}}^i]$
      $\hat{\mathbf{y}}_{t+1}^i, \mathbf{f}_{t+1} \leftarrow \mathcal{F}(\mathbf{x}_{1:t+1}^i)$
      Compute $\mathbf{S}_{0,t+1}^i(\mathbf{x}_{1:t+1}^i)$ and $\mathbf{S}_{1,t+1}^i(\mathbf{x}_{1:t+1}^i)$ using (3)
      if $E(\mathbf{S}_{0,t+1}^i, \mathbf{S}_{1,t+1}^i, \hat{\mathbf{q}}_0^A, \hat{\mathbf{q}}_1^A) = 1$ **or** $t = T-1$ **then**
        /* Collect non-conformity scores at time $t+1$ */
        for $j = 1, 2, \ldots, C$ **do**
          if $y_j^i = 0$ **then**
            $S_0[j] \leftarrow S_0[j] \cup \{S_{0,t+1,j}^i\}$
          else
            $S_1[j] \leftarrow S_1[j] \cup \{S_{1,t+1,j}^i\}$
          end if
        end for
        break
      end if
    end for
  end for
  /* Compute refined per-class quantiles $\hat{\mathbf{q}}_0^B, \hat{\mathbf{q}}_1^B$ */
  $\hat{q}_{y,j}^B = \text{Quantile}\left(S_y[j]; \frac{\lceil (n_{\mathrm{B}_{y,j}}+1)(1-\alpha) \rceil}{n_{\mathrm{B}_{y,j}}}\right),\ y = 0, 1,\ j = 1, 2, \ldots, C.$
  return $\hat{\mathbf{q}}_0^B, \hat{\mathbf{q}}_1^B$
end function

Algorithm 4 Calibrate the decision-making thresholds

function CALIBRATE(transformer classifier $\mathcal{F}$, acquisition policy π, calibration dataset $\mathcal{D}_{\text{calib}} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^{n_{\text{calib}}}$, miscoverage rate α)
  Split $\mathcal{D}_{\text{calib}}$ into $\mathcal{D}_A$ and $\mathcal{D}_B$
  /* Compute quantiles using final time step T */
  $\hat{\mathbf{q}}_0^A, \hat{\mathbf{q}}_1^A \leftarrow$ Calibrate_A($\mathcal{F}, \mathcal{D}_A, \alpha$)
  /* Recalibrate quantiles using policy π */
  $\hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1 \leftarrow$ Calibrate_B($\mathcal{F}, \pi, \mathcal{D}_B, \hat{\mathbf{q}}_0^A, \hat{\mathbf{q}}_1^A, \alpha$)
  return $\hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1$
end function

for the contrastive loss computation. These augmentations include horizontal and vertical flips with probability 0.5, random rotations in the range $[-90, 90]$ degrees, and normalization. Figure 3 shows how many times each defect type occurs in the dataset, illustrating class imbalance and highlighting the need for robust evaluation metrics such as the F1 score.

Models Overall, we utilize two deep learning models that are trained sequentially: first, we pretrain the classifier—a causal Transformer—on random patch sequences; second, we train an RL-based patch selector that optimizes the acquisition process given the pretrained classifier; finally, we learn

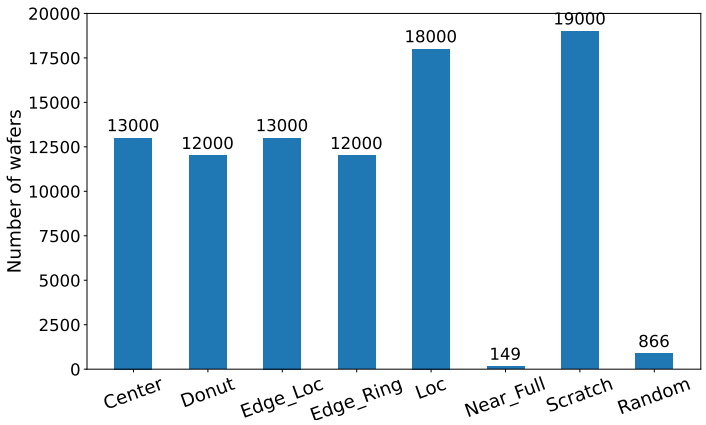


Figure 3: Number of defects of each type present in the MixedWM38 dataset.

the conformal prediction thresholds. Table 1 summarizes how the dataset was split to train each of the components.

Table 1: Number of wafer maps in each subset that are used for training and evaluation.

Dataset	Purpose	# of images
$\mathcal{D}_1$	train transformer	20,528
$\mathcal{D}_2$	learn conformal prediction thresholds $\hat{\mathbf{q}}_0$, $\hat{\mathbf{q}}_1$	2,281
$\mathcal{D}_3$	train patch selector	7,603
$\mathcal{D}_4$	test both transformer and patch selector	7,603

For the Transformer architecture, we employ three different configurations, which we refer to as *small*, *medium*, and *large* based on the number of learnable parameters. Table 2 highlights the differences among these three configurations. Here, `embedding_dim` denotes the dimension of $\mathbf{z}_t$, and L represents the number of Transformer blocks (see Figure 2). Each of the configurations is pretrained for 240 epochs using a learning rate of 0.001. Additionally, we employ KV-caching to speed up the inference stage. More detailed information is available in our code implementation.

To train the RL agent, we utilize the PPO algorithm [46], where both the actor and critic networks are implemented as two-layer MLPs. Additionally, we employ a memory buffer to store transitions and stabilize the learning process. The agent is optimized for 100 iterations using the Adam optimizer with a learning rate of 0.001, the penalty term $c_{\text{RL}} = 0.005$.

Table 2: Difference among the three Transformer configurations: small, medium and large.

Capacity	embedding_dim	L
small	128	8
medium	256	4
large	256	8

As baseline methods, we use SupConWI-RL [16] and its variants. Additionally, we compare our results with the ResNet-34 model [47], a standard convolutional neural network (CNN) baseline. It is worth noting that ResNet-34 requires the full image to be acquired before processing, which makes it less efficient in terms of scanning time. However, it achieves state-of-the-art accuracy on this dataset and serves as an upper-bound benchmark.

Evaluation metrics To validate our model and compare it with others, we focus on Accuracy $= \frac{\#\{i=1,2,\ldots,n_{\text{test}}:\, \mathbf{y}_i=\hat{\mathbf{y}}_i\}}{n_{\text{test}}}$ (therefore, an image is correctly classified only if all components of the output $\hat{\mathbf{y}} \in \{0,1\}^C$ are correctly predicted), F1 $= \frac{2\cdot\text{Recall}\cdot\text{Precision}}{\text{Recall}+\text{Precision}}$, where Recall $= \frac{\text{TP}}{\text{TP+FN}}$ and Precision $= \frac{\text{TP}}{\text{TP+FP}}$ to measure how well the model predicts despite class imbalance. Here, TP, FP,

and FN are the number of true positives, false positives, and false negatives, respectively, and t^* is the number of scanning steps.

Our objective is to evaluate the trade-off between performance (accuracy and F1 score) and the scanning time (t^*) required by different models. CNNs typically achieve state-of-the-art results by utilizing the full wafer surface. Therefore, our goal is to identify a model that achieves comparable or better accuracy/F1 score while significantly reducing the required scanning time.

4.2 Results

First, we compare the LSTM, the proposed Transformer, and CNN models when all information is available, i.e., all patches have been acquired. Table 3 summarizes the test accuracy on the MixedWM38 dataset when a full sequence of measurements is given to the classifier. The Transformer-based architectures achieve higher classification accuracy than the LSTM. It is noteworthy that all three transformer configurations give similar results despite significant differences in the number of parameters. Meanwhile, the CNN outperforms the LSTM and Transformer-based models. Nevertheless, the latter allow for predictions based on partial observations, as we show below. Additionally, the CNN model has significantly more parameters to train and requires more memory to store.

Table 3: Comparison of models in terms of accuracy and number of trainable parameters if all measurements are available to the network. For the Transformer, three different configurations are considered. The results are averaged over five independent runs, providing mean and standard deviation across runs.

Model	Accuracy (%)	# of learnable parameters
LSTM	96.35 ± 0.25	1,508,232
Transformer	98.02 ± 0.14	1,641,160
	98.16 ± 0.06	3,285,384
	98.10 ± 0.11	6,444,424
CNN	98.79 ± 0.09	21,348,168

Table 4 shows the results when applying our adaptive acquisition strategy compared to the SupConWI-RL variants. Our UFLD achieves significantly higher accuracy while maintaining a comparable scanning rate. Moreover, UFLD requires two deep learning models to be trained—the classifier and the RL agent—which is a more parameter-efficient strategy compared to SupConWI-RL, where the confidence estimator is another black-box deep learning model. Instead, conformal prediction provides a simple, fast, and justified approach as a stopping rule. An additional analysis of the effect of the miscoverage rate α is presented in Section 4.3.

Table 4: Comparison of the proposed UFLD with other methods. For single-model cases, the results are averaged over five independent runs, providing mean and standard deviation across runs.

	Model	Accuracy (%)	Scanned patches (%)	# of parameters
CNN	ResNet-34	98.79 ± 0.09	100	21,348,168
Non-hybrid	Single SupConWI-RL	95.58 ± 0.26	59.3 ± 2.6	3,194,554
	Ensemble SupConWI-RL	96.34	60.4	15,972,770
	Single UFLD-large (ours)	97.88 ± 0.07	61.4 ± 1.6	7,174,194
	Ensemble UFLD-small (ours)	97.82	59.7	11,342,650
Hybrid	Single SupConWI-RL	97.74 ± 0.20	59.3 ± 2.6	24,542,722
	Ensemble SupConWI-RL	98.45	60.4	37,320,938
	Single UFLD-large (ours)	98.38 ± 0.10	61.4 ± 1.6	28,522,362
	Ensemble UFLD-small (ours)	98.37	59.7	32,690,818

First, we compare results in the non-hybrid setting (see Remark 3.2), i.e., the case where we do not use CNN even if the whole surface was scanned. UFLD outperforms SupConWI-RL in both single-model and ensemble cases. Moreover, even single UFLD achieves $1.5\,\mathrm{p.p.}$ higher accuracy than Ensemble SupConWI-RL, while having half as many trainable parameters.

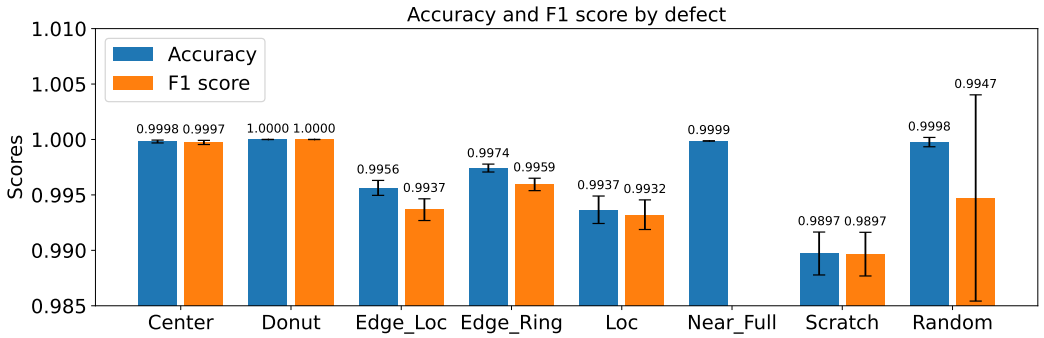


Figure 4: Accuracy and F1 score of UFLD-large calculated by defect and averaged over five runs with 2-sigma error bars.

In the hybrid setting, we do not observe much difference in either accuracy or scanned patches between UFLD variants and Ensemble SupConWI-RL, while the latter requires more trainable parameters. Single SupConWI-RL loses to our UFLD by 0.64 p.p. without significant advantage in scanning time. However, UFLD provides a trade-off between scanning time and accuracy that is controlled by α, which we analyze in Section 4.3. This flexibility allows for an adjustment of the inspection policy based on the specific sensitivity requirements of wafer production.

To analyze our results further, we demonstrate that UFLD effectively handles the class imbalance present in the MixedWM38 dataset. Figure 4 shows the accuracy and F1 score for predicting each defect separately. We can see consistency in both metric values across all defect types, which illustrates the robustness of UFLD despite class imbalance. Notably, we observe a slightly lower F1 score for *Near_Full*, which can be explained by the very small number of these examples in the dataset.

In addition, we compare the overall number of scanned patches between our UFLD (large) and the Ensemble SupConWI-RL in Figure 5. Despite the equal average percentages of scanned surface (i.e., the number of patches), the behavior is different—UFLD has significantly fewer cases where it cannot stop the scanning process early. Ensemble SupConWI-RL more frequently reaches the maximum budget of patches due to higher uncertainty than UFLD. This hints that the conformal prediction framework provides a more robust and better-calibrated confidence threshold than the auxiliary confidence estimator counterpart.

4.3 Sensitivity of conformal prediction to the choice of hyperparameters

We want to show that our approach is robust to the choice of α, regardless of transformer architecture. Table 5 demonstrates how coverage, percentage of scanned regions, and two types of accuracy change with small changes in the miscoverage level α for three variants of UFLD – small, medium and large. The results show a monotonic increase in accuracy and a decrease in coverage with the increase of α; the number of scanned patches also increases with the increase of α. This is consistent with our two-rule conformal stopping criterion: increasing α decreases the conformal thresholds $\hat{\mathbf{q}}_0$ and $\hat{\mathbf{q}}_1$, making positive or negative label certification stricter (see Remark 3.1). As a result, fewer labels are certified at early acquisition steps, and the RL policy must inspect more regions before meeting the stopping condition. This longer acquisition process provides the classifier with more information, which leads to higher accuracy. We observe the same behavior across all three UFLD variants, meaning that the stopping mechanism is stable with respect to model architecture. Here we report the empirical micro-averaged coverage at stopping time:

$$\text{Coverage} = \frac{1}{NC} \sum_{i=1}^{N} \sum_{j=1}^{C} \mathbf{1}\{S_{i,\tau_i,j} \leq \hat{q}_{y_j^i,j}\},$$

where τ_i is the stopping time of wafer i.

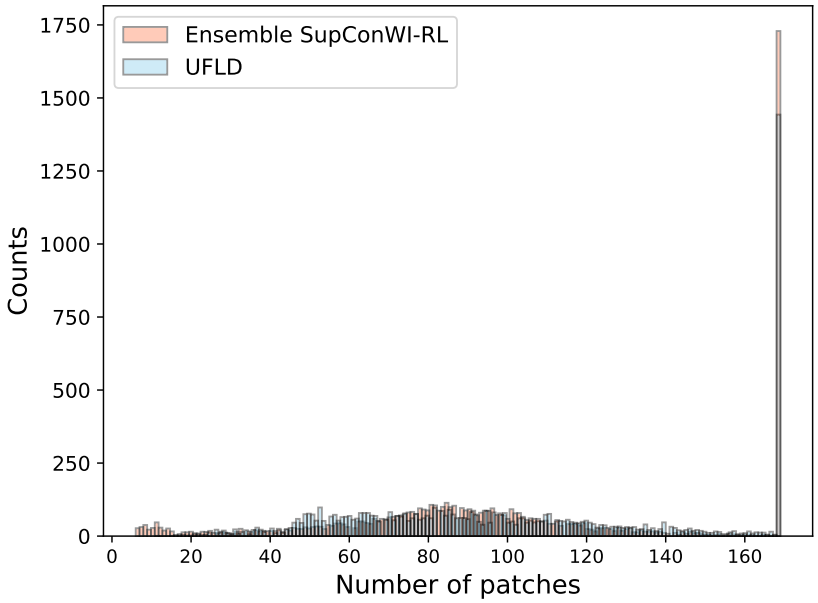


Figure 5: Comparison between Ensemble SupConWI-RL and UFLD (large) in the number of scanned patches.

Table 5: Results of UFLD models of different sizes (1.6M, 3.3M, and 6.4M), varying α from 0.05 to 0.1 with a step size of 0.005.

UFLD		0.05	0.055	0.06	0.065	0.07	0.075	0.08	0.085	0.09	0.095	0.1
small	Non-hybrid Coverage (%)	97.44	97.13	96.74	96.36	95.94	95.50	95.03	94.54	94.04	93.60	93.04
	Scanned (%)	54.62	56.57	58.54	60.21	61.75	63.23	64.51	66.00	67.43	68.55	69.69
	Accuracy (%)	97.34	97.44	97.51	97.56	97.59	97.64	97.67	97.70	97.71	97.72	97.74
	Hybrid Accuracy (%)	97.90	98.06	98.18	98.26	98.31	98.38	98.42	98.46	98.50	98.51	98.55
medium	Non-hybrid Coverage (%)	97.18	96.79	96.48	96.13	95.72	95.30	94.87	94.50	94.04	93.49	93.01
	Scanned (%)	55.45	57.30	58.68	60.21	61.76	63.25	64.54	65.72	67.08	68.42	69.58
	Accuracy (%)	97.58	97.68	97.72	97.73	97.78	97.82	97.83	97.85	97.87	97.87	97.88
	Hybrid Accuracy (%)	98.01	98.13	98.20	98.22	98.28	98.33	98.37	98.40	98.44	98.46	98.47
large	Non-hybrid Coverage (%)	97.34	96.97	96.60	96.25	95.85	95.47	95.05	94.60	94.25	93.83	93.34
	Scanned (%)	56.3	58.3	60.0	61.4	62.9	64.2	65.5	66.9	68.0	69.0	70.1
	Accuracy (%)	97.72	97.80	97.85	97.88	97.91	97.94	97.94	97.96	97.99	98.01	98.03
	Hybrid Accuracy (%)	98.18	98.29	98.34	98.38	98.44	98.48	98.50	98.52	98.55	98.58	98.60

4.4 Ablation study

In addition to the comparison with the baselines, we conduct an ablation study to validate the contribution of each of the components in UFLD. Table 6 summarizes the classification accuracy and F1 score of a Causal Transformer classifier under different configurations. First, we evaluate the classifier that receives 61.5% of the patches using random sequences and RL-based adaptive data acquisition (ADA). This percentage matches the amount of surface that UFLD needs on average for reliable predictions. We perform this comparison to validate the effect of ADA. Second, we consider the configuration where the classifier receives the random sequence of patches but uses the conformal prediction (CP) stopping rule for certainty in the predictions. The third configuration corresponds to the UFLD algorithm, where we utilize both of these techniques.

Overall, we observe a significant performance improvement when we add each of the components: Causal Transformer alone performs the worst with the given data budget, while adding RL-based ADA increases both accuracy ($+4.19$ p.p.) and F1 score ($+0.008$), maintaining the same data budget of 61.5%. The classifier with the CP-based stopping rule without input sequence optimization also improves the indicators ($+4.49$ p.p. for the accuracy and $+0.007$ for the F1 score) but leads to an increase of 9 p.p. in required patches. Finally, the employment of both ADA and the CP-

based stopping rule boosts both accuracy ($+5.08$ p.p.) and the F1 score ($+0.008$) utilizing the same percentage of the wafer surface (61.4%).

Table 6: Ablation study results where the Transformer classifier (large) with or without adaptive data acquisition (ADA) and conformal prediction (CP) stopping rule are considered. The results are averaged over five independent runs.

Model	RL	CP	Acc (%)	F1	Scanned (%)
Transformer	no	no	92.80	0.985	61.5
+ ADA	yes	no	96.99	0.993	61.5
+ Stopping rule	no	yes	97.29	0.992	70.5
UFLD	yes	yes	97.88	0.993	61.4

5 Conclusion

In this work, we addressed the wafer map defect classification problem using an adaptive data acquisition approach. We propose UFLD, a reinforcement learning-based algorithm that adaptively selects the next location for the measurement. In order to make the acquisition process more efficient and reduce operational costs, we employ a conformal prediction-based stopping rule. In contrast to previous work in the semiconductor domain, which requires an additional black-box deep learning model for confidence estimation, the conformal prediction framework is more parameter-efficient. In addition, the proposed UFLD uses a causal transformer architecture as a classifier, instead of RNN-based architectures that were utilized in previous works. This results in a 1.5 p.p. increase in diagnostic accuracy, requiring around 60% of the surface, which is the same number of measurements as previous methods. Moreover, single UFLD uses 50% fewer parameters compared to the baselines due to the elimination of the auxiliary confidence estimator network. Therefore, it further closes the gap to CNN-based approaches that require full scanning. Additionally, variations in the miscoverage rate provide a controllable trade-off between inspection performance and scanning time.

In our opinion, the UFLD framework contributes to more reliable and cost-efficient semiconductor manufacturing. By reducing the need for full surface scanning, our approach increases the throughput of inspection tools in high-volume manufacturing settings. Furthermore, conformal prediction provides a principled mechanism for constructing prediction sets per label and defining the stopping criterion through an interpretable miscoverage parameter α, compared to black-box deep learning methods. Future work could improve the current framework by developing more rigorous theoretical guarantees and by exploring extensions of this adaptive acquisition strategy to other problems, such as defect segmentation or surface reconstruction.

Acknowledgments and Disclosure of Funding

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] Gordon E Moore. Cramming more components onto integrated circuits. *Electronics*, 38, 1965.

[2] Victor Azamfirei, Foivos Psarommatis, and Yvonne Lagrosen. Application of automation for in-line quality inspection, a zero-defect manufacturing approach. *Journal of Manufacturing Systems*, 67:1–22, 2023.

[3] Jay Lee, Hossein Davari, Jaskaran Singh, and Vibhor Pandhare. Industrial artificial intelligence for industry 4.0-based manufacturing systems. *Manufacturing letters*, 18:20–23, 2018.

[4] Paul Sandland. Automated defect inspection: past, present, and future. In *Metrology, Inspection, and Process Control for Microlithography XII*, volume 3332, pages 296–308. SPIE, 1998.

[5] Mira Park, Jesse S Jin, Sherlock L Au, Suhuai Luo, and Yue Cui. Automated defect inspection systems by pattern recognition. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2(2):31–42, 2009.

[6] Rahul Rai, Manoj Kumar Tiwari, Dmitry Ivanov, and Alexandre Dolgui. Machine learning in manufacturing and industry 4.0 applications. *International Journal of Production Research*, 59(16):4773–4778, 2021.

[7] Farhan A Alenizi, Shirin Abbasi, Adil Hussein Mohammed, and Amir Masoud Rahmani. The artificial intelligence technologies in industry 4.0: A taxonomy, approaches, and future directions. *Computers & Industrial Engineering*, 185:109662, 2023.

[8] Guizhong Fu, Peize Sun, Wenbin Zhu, Jiangxin Yang, Yanlong Cao, Michael Ying Yang, and Yanpeng Cao. A deep-learning-based approach for fast and robust steel surface defects classification. *Optics and Lasers in Engineering*, 121:397–405, 2019.

[9] Swarit Anand Singh and Kaushal A Desai. Automated surface defect detection framework using machine vision and convolutional neural networks. *Journal of Intelligent Manufacturing*, 34(4):1995–2011, 2023.

[10] Taiheng Liu, Zhaoshui He, Zhijie Lin, Guang-Zhong Cao, Wenqing Su, and Shengli Xie. An adaptive image segmentation network for surface defect detection. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):8510–8523, 2022.

[11] Aleksandr Dekhovich and Miguel A Bessa. Continual learning for surface defect segmentation by subnetwork creation and selection. *Journal of Intelligent Manufacturing*, 36(5):3051–3065, 2025.

[12] Yuxin Ma, Jiaxing Yin, Feng Huang, and Qipeng Li. Surface defect inspection of industrial products with object detection deep networks: A systematic review. *Artificial Intelligence Review*, 57(12):333, 2024.

[13] Wei Zhu, Hui Zhang, Chao Zhang, Xiaoyang Zhu, Zhen Guan, and Jiale Jia. Surface defect detection and classification of steel using an efficient swin transformer. *Advanced Engineering Informatics*, 57:102061, 2023.

[14] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

[15] Aleksandr Dekhovich, Oleg Soloviev, and Michel Verhaegen. Sequential wafer map inspection via feedback loop with reinforcement learning. *Expert Systems with Applications*, 275:126996, 2025.

[16] Aleksandr Dekhovich and Oleg Soloviev. Supconwi-rl: Wafer inspection with reinforcement learning enhanced by supervised contrastive learning. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1396–1405. IEEE, 2025.

[17] 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.

[18] Naigong Yu, Huaisheng Chen, Qiao Xu, Mohammad Mehedi Hasan, and Ouattara Sie. Wafer map defect patterns classification based on a lightweight network and data augmentation. *CAAI Transactions on Intelligence Technology*, 8(3):1029–1042, 2023.

[19] Wei Yuan, Jinda Yan, and Minghao Piao. Position-aware self-supervised learning for wafer map defect pattern recognition. *IEEE Transactions on Instrumentation and Measurement*, 2025.

[20] Han Kyul Kim and Jaewoong Shim. Generalized zero-shot learning for classifying unseen wafer map patterns. *Engineering Applications of Artificial Intelligence*, 133:108476, 2024.

[21] Hao Geng, Fan Yang, Xuan Zeng, and Bei Yu. When wafer failure pattern classification meets few-shot learning and self-supervised learning. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–8. IEEE, 2021.

[22] Jianbo Yu, Zongli Shen, and Shijin Wang. Wafer map defect recognition based on deep transfer learning-based densely connected convolutional network and deep forest. *Engineering Applications of Artificial Intelligence*, 105:104387, 2021.

[23] Jia-Hong Chou, Fu-Kwun Wang, and Hsuan-Kai Chen. Adaptive transfer learning with deformable convolution for wafer defect pattern recognition. *IEEE Transactions on Semiconductor Manufacturing*, 2025.

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

[25] Tongwha Kim and Kamran Behdinan. Advances in machine learning and deep learning applications towards wafer map defect recognition and classification: a review. *Journal of Intelligent Manufacturing*, 34(8):3215–3247, 2023.

[26] Jaewoong Shim, Seokho Kang, and Sungzoon Cho. Active cluster annotation for wafer map pattern classification in semiconductor manufacturing. *Expert Systems with Applications*, 183:115429, 2021.

[27] Shu-Kai S Fan and Shang-Hao Chiu. A new vit-based augmentation framework for wafer map defect classification to enhance the resilience of semiconductor supply chains. *International Journal of Production Economics*, 273:109275, 2024.

[28] Shaurya Jain, A Charan Kumari, and K Srinivas. Sdcvit: a transformer-powered hierarchical framework for semiconductor defect detection and classification. *Journal of Intelligent Manufacturing*, pages 1–31, 2026.

[29] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[30] Samantha Lafrance and Shonda Bernadin. Evaluating visual transformers for wafer defect detection: A feasibility study. In *SoutheastCon 2025*, pages 1288–1293. IEEE, 2025.

[31] Faisal Mohammad and Duksan Ryu. Semiconductor wafer map defect classification with tiny vision transformers. *arXiv preprint arXiv:2504.02494*, 2025.

[32] Shouhong Chen, Zhentao Huang, Tao Wang, Xingna Hou, and Jun Ma. Wafer map defect recognition based on multi-scale feature fusion and attention spatial pyramid pooling. *Journal of Intelligent Manufacturing*, 36(1):271–284, 2025.

[33] Xinting Liao, Jie Zhang, Youlong Lyu, and Junliang Wang. Wafer defect semantic reasoning via a three-stage retrieval-augmented system in semiconductor manufacturing. *Journal of Manufacturing Systems*, 85:269–286, 2026.

[34] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[35] Cassandra Tong Ye, Jiashu Han, Kunzan Liu, Anastasios Angelopoulos, Linda Griffith, Kristina Monakhova, and Sixian You. Learned, uncertainty-driven adaptive acquisition for photon-efficient scanning microscopy. *Optics Express*, 33(6):12269–12287, 2025.

[36] Elizabeth A Holman, Yuan-Sheng Fang, Liang Chen, Michael DeWeese, Hoi-Ying N Holman, and Paul W Sternberg. Autonomous adaptive data acquisition for scanning hyperspectral imaging. *Communications biology*, 3(1):684, 2020.

[37] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, Cambridge, 1998.

[38] Jean-Raymond Betterton, Daniel Ratner, Samuel Webb, and Mykel Kochenderfer. Reinforcement learning for adaptive illumination with x-rays. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 328–334. IEEE, 2020.

[39] 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.

[40] 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.

[41] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of machine learning research*, 9(3), 2008.

[42] Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in neural information processing systems*, 33:3581–3591, 2020.

[43] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*, 2020.

[44] Anastasios N Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2022.

[45] 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.

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

[47] 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.