MaskVision: A Novel Open EUV Semiconductor Photomask Benchmark for Anomaly Detection in Semiconductor Manufacturing

Gibeom Kim[1], Zaigham Zaheer[2], Seongjin Choi[3], Karthik Nandakumar[4] and Hyejin S Kim[5]

***Abstract*— We present a large scale EUV (Extreme Ultra Violet) Semiconductor Photomask dataset, MaskVision, for anomaly detection. We also introduce a training free DNN method called Vector Binning Anomaly Detection (VBAD), which emphasizes speed and accuracy in industrial contexts. The MaskVision dataset features large scale images carefully curated to reflect real world semiconductor manufacturing conditions, enabling thorough evaluation of scalability and robustness. In industrial contexts, high resolution optical systems are used to identify small defects that are low in contrast and difficult to distinguish. Existing datasets rarely address this issue, motivating our novel MaskVision dataset. MaskVision provides 45 times more training data and 41 times more test data per class than MVTec-AD, the largest existing benchmark for industrial anomaly detection. Our VBAD approach leverages pretrained network embeddings and a lightweight binning mechanism to eliminate time consuming network training, thereby reducing memory usage and inference time. The proposed approach reduces computational load by computing only binning and channel-wise transitions. Although conventional methods evaluate channel, height, and width simultaneously, resulting in high complexity, our method exploits the observation that, for normal data, transition differences are similar across channels and spatial positions. By calculating only these transitions, we significantly reduce computations and accelerate processing. Experiments on the existing benchmark dataset MVTec-AD and our proposed MaskVision dataset demonstrate that VBAD achieves competitive detection and segmentation performance while having significantly lower computational overhead compared to conventional deep learning based methods. These findings underscore the feasibility of VBAD for real time anomaly inspection in high volume scenarios; for example, on a workload of around 100,000 images, VBAD runs approximately 4–6× faster than methods such as SimpleNet. MaskVision dataset can be downloaded at [Link].**

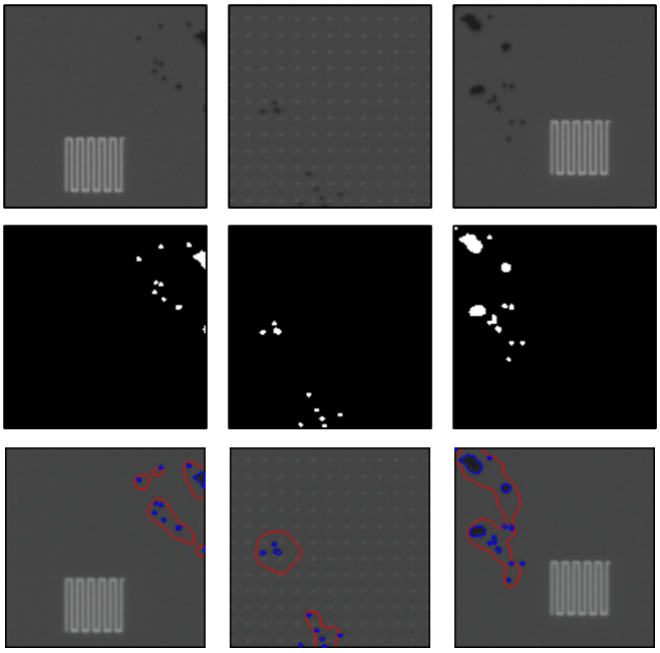

Fig. 1. The top row shows the original images of MaskVision, the middle row shows the ground truth defect masks, and the bottom row overlays the VBAD model's predicted defect regions (in red) along with the ground truth regions (in blue).

I. INTRODUCTION

Visual anomaly detection plays a crucial role in numerous fields, including industrial inspection [5], medical imaging [23], and autonomous driving [15]. Although deep learning based methods have achieved remarkable accuracy, their substantial training effort and extensive computational requirements often limit scalability in real world production environments. This challenge becomes especially acute when inspecting large volumes of data, as is often common in semiconductor manufacturing, where traditional methods risk memory overflows or prohibitively slow inference times.

In many industrial applications, detecting small defects is critical for product quality, but creating large scale datasets remains expensive. Advanced optical inspection systems are required to capture such detailed images, and Their cost and the complexity of processing massive image volumes often deter organizations from publicly releasing data. As a result, most existing datasets prioritize accuracy on moderate sized or lower resolution images, overlooking the practical constraints of real world deployments including high resolution data and real time requirements.

To address this gap, we introduce a new EUV (Extreme Ultra Violet) Semiconductor Photomask dataset, MaskVision. The images in this dataset are from semiconductor lithography processes, in which even minor defects can significantly impact product yield. By publicly sharing this dataset, we aim to establish an industrial scale benchmark

[1]Gibeom Kim is with Faculty of Artificial Intelligence, University of Science & Technology, Yuseong-gu, Daejeon, Republic of Korea chroion0@gmail.com

[2]Zaigham Zaheer is with Mohamed bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi, United Arab Emirates zaigham.zaheer@mbzuai.ac.ae

[3]Seongjin Choi is with FINE SEMITECH CORP, Hwaseong-si, Gyeonggi-do, Korea sjchoi@fstc.co.kr

[4]Karthik Nandakumar is with Mohamed bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi, United Arab Emirates karthik.nandakumar@mbzuai.ac.ae

[5]Hyejin S Kim is with Electronics and Telecommunications Research Institute, Yuseong-gu, Daejeon, Republic of Korea marisan@etri.re.kr

that challenges anomaly detection algorithms to excel in both speed and accuracy, filling a crucial void in the availability of large scale anomaly detection data.

To bridge the gap between the existing literature and deployable industrial solutions for anomaly detection, we also propose **Vector Binning Anomaly Detection (VBAD)**. VBAD is designed to minimize both computational overhead and memory usage by eliminating the need for DNN training. Unlike prior methods that simultaneously analyze the dimensions *channel*, *height*, and *width* of a feature map, which cause complexity to grow cubically with spatial resolution, VBAD performs a lightweight *binning* step and then models *only channel-wise transitions*.

This design is motivated by our empirical observation that, transition statistics are almost invariant across spatial positions. Therefore, aggregating over height and width does not degrade detection ability but yields a drastic reduction in computations. In practice, the proposed strategy replaces millions of three-dimensional transition counts with a compact set of one-dimensional channel transitions, providing orders of magnitude speed ups while preserving accuracy.

We evaluated VBAD on our MaskVision dataset and on the widely used MVTec-AD benchmark [4] to cover both standard and large scale scenarios. Experimental results show that VBAD's accuracy is comparable to state-of-the-art methods, while its inference runs significantly faster and more scalable. Notably, the MaskVision dataset reveals the limitations of training intensive approaches, which often encounter memory overflows on substantial, high resolution image sets. In contrast, VBAD is designed to meet industrial scale demands with only a minimal trade-off in accuracy.

Overall, this paper offers two key contributions:

- We propose a *DNN training free* based anomaly detection framework (VBAD) that efficiently localizes defects via compact channel-wise transition matrices, achieving low computational complexity and scalability on large datasets with competitive accuracy.
- We release a new EUV Semiconductor Photomask dataset, **MaskVision**, featuring challenging, high resolution images that serve as a rigorous benchmark for real world, industrial scale anomaly detection research.

By addressing both computational efficiency and large scale data handling, our work paves the way for more practical and precise inspection systems, which are increasingly critical need in advanced manufacturing settings.

II. Related Works

Visual anomaly detection is a critical area of research with various applications, such as medical imaging, autonomous driving, and industrial inspection [15]. This section outlines some key methodologies and approaches.

A. Reconstruction Based Methods

Reconstruction based approaches [14], [26], [20] typically involve training deep models (e.g., autoencoders [3], [4], [9], [21]) to reconstruct input images. These models are expected to accurately reconstruct normal data while failing to do so for anomalous data. The difference between an original and its corresponding reconstructed image serves as the basis for detecting anomalies. However, this approach can be limited by the quality of the reconstruction [2], particularly when subtle defects are difficult to capture.

B. Embedding Based Methods

These methods leverage pretrained neural networks to extract and compress features into a compact space [7], [13]. Anomalous data, which do not conform to the learned normal patterns, stand out as outliers in this feature space. Techniques such as memory banks [11], [16], [22] store representative normal feature vectors, which are then compared to new data to identify anomalies through metric learning. One-class classification methods further refine [1] this by establishing explicit boundaries around normal data clusters, such as hyperplanes [18] or hyperspheres [17].

C. Anomaly Detection Datasets

A variety of public datasets exist for benchmarking anomaly detection, the most common being MVTec-AD [4] and VisA [24], each providing thousands of normal and anomalous images with pixel level annotations. Kolektor SDD [19] is another notable industrial dataset, featuring electrical component images under diverse conditions. While these datasets have significantly advanced research in anomaly detection, their relatively moderate resolutions often limit their applicability to micro scale defect inspection. In many industrial scenarios such as semiconductor manufacturing, even weak defects can critically impact product yield, so high resolution images are required for defect detection. This requirement exceeds the capabilities of most existing public datasets and highlights the need for larger, more complex benchmarks that can properly evaluate both accuracy and scalability in real world, high resolution anomaly detection tasks.

III. Method

In this section, we introduce our anomaly detection approach, which is designed to detect anomalies efficiently while maintaining low computational complexity. The method consists of three key stages: *feature extraction*, *transition-matrix calculation*, and *distance-matrix calculation*, ultimately producing an anomaly map that highlights defective regions in an input image.

A. Feature Extraction

Given an input image I of size $H \times W$, we first extract feature embeddings using a pretrained convolutional neural network Φ [1], [6], [8]. Each image passes through Φ, and we collect feature maps from a specific intermediate layer. These feature maps are referred to as embedding vectors E, where each vector corresponds to a patch of the input image. Thus, we represent the embedding vector at location (X_i, Y_j) as $E_{i,j} \in \mathbb{R}^C$, where i, j index the spatial locations within the feature map.

To enhance the time efficiency of our anomaly detection method, we applied a binning technique to the embedding

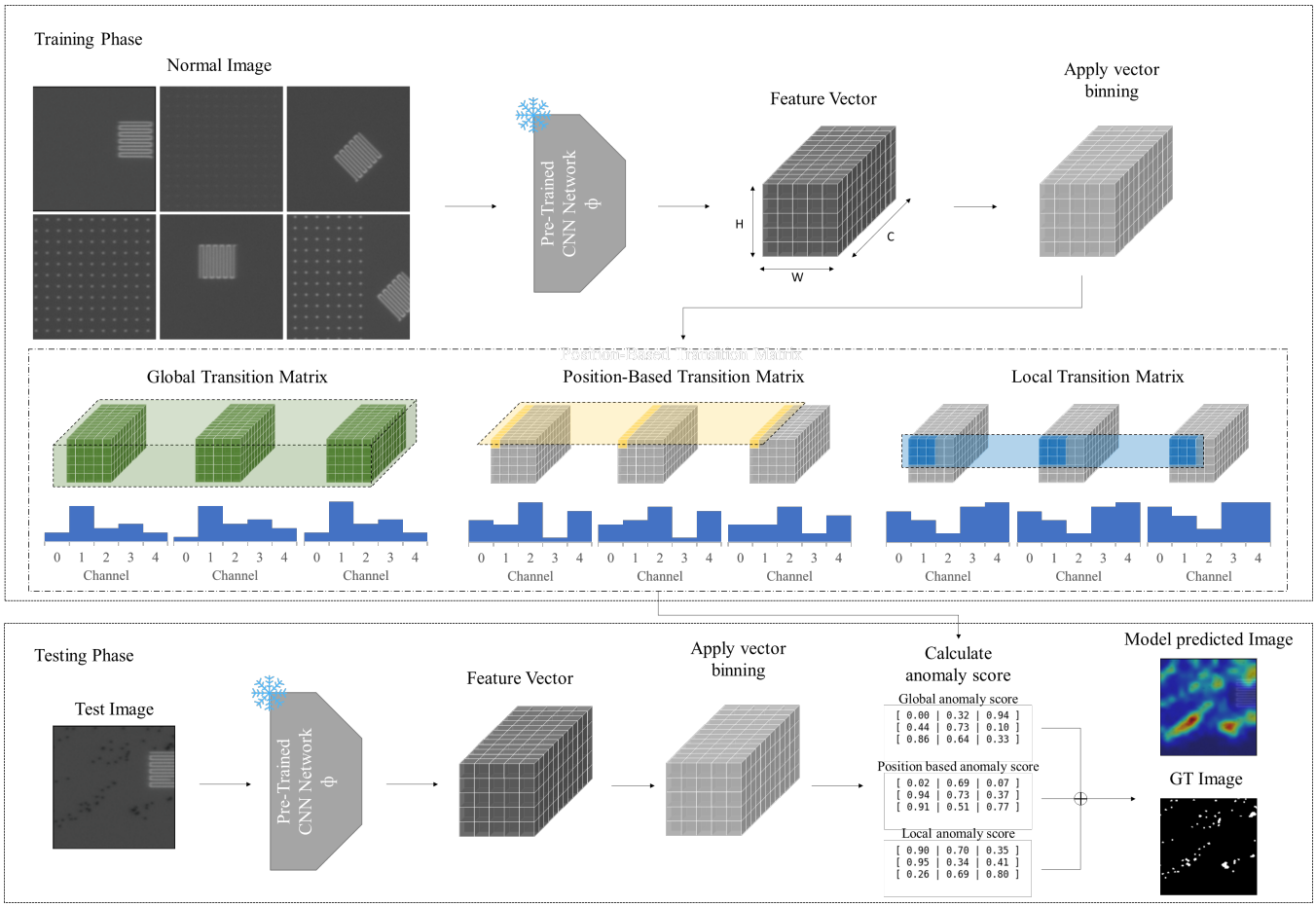

Fig. 2. VBAD Pipeline (Overview). Six normal images represent sample images of MaskVision. And these input training images are passed through a frozen, pre-trained CNN (Φ) to extract multi-channel feature embeddings. Each channel's continuous values are discretized via vector binning into K states, and three transition matrices are learned from these discrete states: (1) global (green) matrices capturing state transitions across channels of all vectors, (2) position-based (yellow) matrices capturing transitions at each spatial location across channels, and (3) local (blue) matrices capturing transitions between neighboring spatial positions within each channel. Below each matrix block, the blue histograms show how the same binned feature vectors yield different count distributions depending on the transition counting scheme. At test time, a new image's embeddings are binned and scored against these matrices to produce three distance matrices, the sum of which forms an anomaly map highlighting deviations from normal transitions.

vectors. Binning helps to reduce the dimensionality of the feature space and allows for more efficient computation of transition matrices.We apply binning to group the continuous values in each channel into discrete bins. Let b_k denote the bin value for the k-th bin. The binning process for each element $E_{i,j}$ of the embedding vector in channel c is defined as:

$$B_{i,j} \mapsto b_k \quad \text{if} \quad b_k \leq E_{i,j} < b_{k+1} \tag{1}$$

$B_{i,j}$ is discrete vector as location (X_i, Y_j). This process maps each element of the embedding vector to a discrete bin, reducing the continuous feature space to a finite set of states. The number of bins K is a hyperparameter that controls the granularity of the binning process.

After binning, each embedding vector $E_{i,j}$ is represented by a discrete vector $B_{i,j} \in \{b_1, b_2, \ldots, b_K\} \in \mathbb{R}^C$. Next, we calculate the transition counts $N_{b_1,b_2}^{(c,c+1)}$ used in the transition matrices. The count $N_{b_1,b_2}^{(c,c+1)}$ represents the number of transitions observed from b_1 to b_2 in channel c to in channel $c+1$, across all spatial locations (i, j):

$$N_{b_1,b_2,i,j}^{(c,c+1)} = \sum_{m=1}^{M} \delta((B_{m,c,i,j} = b_1) \wedge (B_{m,c+1,i,j} = b_2)) \tag{2}$$

Here, $\delta(\cdot)$ is the Kronecker delta function, which equals 1 when the condition is true and 0 otherwise, and M is the number of images in the dataset. The summation is performed over all spatial positions (i, j) in the feature map. The count $N_{b_1,b_2,i,j}^{(c,c+1)}$ is then used to compute the transition matrices.

B. Transition Matrix Calculation

To model normal feature space behavior at multiple granularities, we build three complementary transition matrices—global, position-based, and local—each capturing a different aspect of channel-wise state transition.

The global transition matrix is computed by pooling state transitions across every spatial location, so its statistics are shift-invariant; even when the object of interest is

translated within the field of view, the aggregate channel to channel patterns remain consistent and allow normal versus abnormal samples to be distinguished robustly under misalignment.

The position-based transition matrix keeps a separate set of transition probabilities for every pixel coordinate (i,j). When images are well aligned, a specific part of the object is expected to recur at that coordinate, so this matrix encodes location specific regularities and highlights deviations that occur only when a particular structural element differs from its nominal appearance.

The local transition matrix tallies transitions between neighboring pixels inside the same channel, thereby capturing the short range transition of feature vectors. It effectively memorizes the object's fine contours and surface texture, so localized defects such as scratches or missing pattern segments appear as unlikely local transitions.

By fusing scores derived from these three matrices, VBAD is simultaneously robust to global shifts, sensitive to position locked variations, and responsive to minute local irregularities, thus achieving precise and reliable anomaly detection in high-resolution industrial imagery.

1) Global Transition Matrix: The global transition matrix captures the transitions between states across different channels over all spatial positions. The transition probability P from from bin b_1 in channel c to bin b_2 in channel $c+1$, aggregated over all spatial positions, is given by:

$$P^{glb}_{c,b1,b2} = \frac{\sum_{x=i}^{X}\sum_{y=j}^{Y} N^{(c,c+1)}_{b_1,b_2,i,j}}{\sum_{x=i}^{X}\sum_{y=j}^{Y}\sum_{p=b_1}^{b_K}\sum_{q=b_1}^{b_K} N^{(c,c+1)}_{p,q,i,j}} \tag{3}$$

$$P^{glb}_{c} = \begin{bmatrix} P_{c,b_1,b_1} & P_{c,b_1,b_2} & \cdots & P_{c,b_1,b_k} \\ P_{c,b_2,b_1} & P_{c,b_2,b_2} & \cdots & P_{c,b_2,b_k} \\ \vdots & \vdots & \ddots & \vdots \\ P_{c,b_k,b_1} & P_{c,b_k,b_2} & \cdots & P_{c,b_k,b_k} \end{bmatrix} \tag{4}$$

Here, $P^{glb}_{c,b1,b2}$ represents the probability of transitions from bin b_1 in channel c to bin b_2 in channel $c+1$ across all positions i,j. The summation over all i,j ensures that the transition matrix accounts for global patterns across the entire feature map. Finally, P^{glb}_{c} is calculated as Global Transition Matrix in channel c. So total global transition matrix is $P^{glb} \in \mathbb{R}^{(C-1)\times K\times K}$.

2) Position-Based Transition Matrix: The position-based transition matrix refines the global transition matrix by considering transitions at specific spatial positions. At a specific position (i,j), the transition probability from state i in channel c to state j in channel $c+1$ is defined as:

$$P^{pos}_{c,b1,b2,i,j} = \frac{N^{(c,c+1)}_{b_1,b_2,i,j}}{\sum_{p=b_1}^{b_K}\sum_{q=b_1}^{b_K} N^{(c,c+1)}_{p,q,i,j}} \tag{5}$$

$$P^{pos}_{c,i,j} = \begin{bmatrix} P^{pos}_{c,b_1,b_1,i,j} & P^{pos}_{c,b_1,b_2,i,j} & \cdots & P^{pos}_{c,b_1,b_k,i,j} \\ P^{pos}_{c,b_2,b_1,i,j} & P^{pos}_{c,b_2,b_2,i,j} & \cdots & P^{pos}_{c,b_2,b_k,i,j} \\ \vdots & \vdots & \ddots & \vdots \\ P^{pos}_{c,b_k,b_1,i,j} & P^{pos}_{c,b_k,b_2,i,j} & \cdots & P^{pos}_{c,b_k,b_k,i,j} \end{bmatrix} \tag{6}$$

$$P^{pos}_{c} = \begin{bmatrix} P^{pos}_{c,1,1} & P^{pos}_{c,1,2} & \cdots & P^{pos}_{c,1,Y} \\ P^{pos}_{c,2,1} & P^{pos}_{c,2,2} & \cdots & P^{pos}_{c,2,Y} \\ \vdots & \vdots & \ddots & \vdots \\ P^{pos}_{c,X,1} & P^{pos}_{c,X,2} & \cdots & P^{pos}_{c,X,Y} \end{bmatrix} \tag{7}$$

In this case, $P^{pos}_{c,b1,b2,i,j}$ represents the probability of position-based transitions from bin b_1 in channel c to bin b_2 in channel $c+1$ at poss (i,j). Unlike the global matrix, this transition matrix is normalized at each spatial position (i,j), allowing the model to detect anomalies that manifest as spatially localized deviations. Finally, $P^{pos}_{c,i,j}$ is calculated as Position-based Transition Matrix in channel c at positions (i,j). So total position-based transition matrix is $P^{pos} \in \mathbb{R}^{(C-1)\times K\times K\times H\times W}$.

3) Local Transition Matrix: The local transition matrix focuses on transitions within the same channel but across neighboring spatial positions. At a given position (i,j) in channel c, the matrix models transition between the position at (i,j) and the neighboring positions within a tolerance $\pm t$. Let Δi and Δj denote the relative offsets from the central position (i,j) within the tolerance t. So, to calculate the local transition matrix, we count discrete vector $B_{i,j}$ as a different method.

$$N^{c}_{b_1,b_2,i,j} = \sum_{m=1}^{M}\sum_{\Delta i=-t}^{t}\sum_{\Delta j=-t}^{t} \delta((B_{m,c,i,j}=b_1) \wedge (B_{m,c,i+\Delta i,j+\Delta j}=b_2)) \tag{8}$$

$$P^{loc}_{c,b1,b2} = \frac{\sum_{x=i}^{X}\sum_{y=j}^{Y} N^{c}_{b_1,b_2,i,j}}{\sum_{x=i}^{X}\sum_{y=j}^{Y}\sum_{p=b_1}^{b_K}\sum_{q=b_1}^{b_K} N^{c}_{p,q,i,j}} \tag{9}$$

$$P^{loc}_{c} = \begin{bmatrix} P^{loc}_{c,b_1,b_1} & P^{loc}_{c,b_1,b_2} & \cdots & P^{loc}_{c,b_1,b_k} \\ P^{loc}_{c,b_2,b_1} & P^{loc}_{c,b_2,b_2} & \cdots & P^{loc}_{c,b_2,b_k} \\ \vdots & \vdots & \ddots & \vdots \\ P^{loc}_{c,b_k,b_1} & P^{loc}_{c,b_k,b_2} & \cdots & P^{loc}_{c,b_k,b_k} \end{bmatrix} \tag{10}$$

C. Distance Matrix Calculation

Here, $P^{loc}_{c,b1,b2}$ represents the probability of transitions from bin b_1 in channel c to bin b_2 in channel $c+1$ across all positions i,j. The summation over all i,j ensures that the transition matrix accounts for global patterns across the entire feature map. Finally, P^{loc}_{c} is calculated as Local Transition Matrix in channel c. So local global transition matrix is $P^{loc} \in \mathbb{R}^{C\times K\times K}$. By structuring the transition matrices in this way, we can effectively model both global and local dependencies, providing a robust mechanism for detecting anomalies in images.

Once the transition matrices are established, the next crucial step is calculating the distance matrix, which quantifies deviations from the normal patterns captured by the transition matrices. These distance matrices are essential for generating the final anomaly score map.

Given an input image, the embedding vector $E_{i,j}$ is discretized into $B_{i,j}$ using the predefined binning process. The

$B_{i,j}$ corresponding to each embedding vector is then used to compute three types of distance matrices: Global, Position-Based, and Local. These matrices account for transitions across different channels (global transitions), specific spatial positions (position-based transitions), and neighboring spatial positions within the same channel (local transitions). The global transition matrix captures transitions between states across different channels over all spatial positions. The global distance matrix $D^{glb}(i,j)$ is calculated by computing the log probability of these transitions, defined as:

$$D_{i,j}^{glb} = \sum_{c=1}^{C-1} |\log(P_c^{glb}(B_{c,i,j}, B_{c+1,i,j}))| \tag{11}$$

where $P_c^{glb}(B_{c,i,j}, B_{c+1,i,j})$ is the transition probability calculated by using discrete vector at c-th channel and (i,j) positions. And by summation all channel's $P_c^{glb}(B_{c,i,j}, B_{c+1,i,j})$, calculate global distance value at position (i,j), denote as $D_{i,j}^{glb}$. The position-based transition matrix refines the global transition matrix by focusing on transitions at specific spatial positions. The position-based distance matrix $D_{i,j}^{pos}$ is computed as:

$$D_{i,j}^{pos} = \sum_{c=1}^{C-1} |\log(P_{c,i,j}^{pos}(B_{c,i,j}, B_{c+1,i,j}))| \tag{12}$$

where $P_{c,i,j}^{pos}(B_{c,i,j}, B_{c+1,i,j})$ is the transition probability calculated by using discrete vector at c-th channel and (i,j) positions. And by summation all channel's $P_{c,i,j}^{pos}(B_{c,i,j}, B_{c+1,i,j})$, calculate position-based distance value at position (i,j), denote as $D_{i,j}^{pos}$. The local transition matrix captures transitions within the same channel but across neighboring spatial positions. The local distance matrix $D_{i,j}^{loc}$ is defined by the log probability of these local transitions:

$$D_{i,j}^{loc} = \sum_{c=1}^{C} \sum_{\Delta i=-t}^{t} \sum_{\Delta j=-t}^{t} |\log(P_c^{loc}(B_{c,i,j}, B_{c,i+\Delta i,j+\Delta j}))| \tag{13}$$

where $P_c^{loc}(B_{c,i,j}, B_{c,i+\Delta i,j+\Delta j})$ is the transition probability calculated by using discrete vector at c-th channel and (i,j) positions. And by summation all channel's $P_c^{loc}(B_{c,i,j}, B_{c,i+\Delta i,j+\Delta j})$, calculate global distance value at position (i,j), denote as $D_{i,j}^{loc}$. The overall distance matrix $D_{i,j}$ for each pixel is obtained by aggregating the contributions from all three distance matrices:

$$D_{i,j} = D_{i,j}^{glb} + D_{i,j}^{pos} + D_{i,j}^{loc} \tag{14}$$

This aggregation ensures that the method captures both global and local deviations effectively. The final anomaly score map is generated by summing the distance matrices across all channels and spatial positions, with higher scores indicating stronger deviations from normal patterns. By integrating information from the global, position-based, and local transitions, the proposed method provides a comprehensive mechanism for detecting anomalies, accounting for both broad and localized deviations within the large scale image data.

D. Lightweight computation

Most anomaly detection models pay a *quadratic* price in the channel dimension. Methods that model feature distributions by retaining a pixel $C \times C$ covariance matrix or an equivalent memory bank of all training embeddings incur an $O(C^2HW)$ memory and FLOP footprint. End-to-end detection models which is based on DNN training, perform $s^2 C_{\text{in}} C_{\text{out}} HW$ multiply–accumulate operations for each convolution layer; when $C_{\text{in}} \simeq C_{\text{out}} = C$, this likewise scales as $O(s^2C^2HW)$, where s denotes the filter size.

VBAD breaks this quadratic barrier by limiting the statistics gathered in *different* transition matrices:

1) **Global & position-based matrices** — we count only the adjacent-channel transition $(c \to c+1)$ instead of all C^2 channel pairs;
2) **Local matrix** — within each channel we tally a fixed $(2t+1)^2$ neighbourhood in the spatial plane, where t is a small constant.

Both statistics are accumulated over the entire $H \times W$ grid, so each pixel triggers only a constant size lookup and increment. The dominant complexity therefore falls to

$$\underbrace{O(CHW)}_{\text{global}} + \underbrace{O(CHW)}_{\text{position-based}} + \underbrace{O\big(t^2CHW\big)}_{\text{local}} \simeq O(CHW), \tag{15}$$

Thus VBAD replaces the C^2 term with a linear C, achieving more than a C-fold reduction compare to other anomaly detection models. This drop in complexity, from $O(C^2HW)$ to $O(CHW)$, is the main reason for the low computational complexity.

In addition to reducing asymptotic complexity, VBAD also benefits from using mainly integer operations during transition counting. Classical methods rely heavily on floating point multiplications and additions to build covariance matrices or memory banks. In contrast, VBAD's core transaction is an integer increment based on discretized bins, which is significantly cheaper than a floating point MAC. Only in a final normalization step are these counts converted to probabilities and processed with floating point arithmetic. Consequently, most of the VBAD workload is executed with integer operations, which provides additional practical speed ups on hardware where integer arithmetic has lower latency and power consumption than floating point. The result of these efficiency improvement is reported in V-B.

IV. MaskVision Dataset

To evaluate the performance and scalability of our proposed method, we employ MaskVision dataset. These images capture the complexities found in semiconductor inspection, including small manufacturing defects, which are approximately 0.7 micrometers size, and naturally occurring anomalies such as dust or contamination. We use slightly low magnification bright field optical setup because of data size increasing, processing times and costs. Therefore, a slightly lower magnification bright field optical setup was chosen for the collection of this dataset to reduce the overall image

dimensions, accepting a minor trade off in locational and shape precision.

Resolution and Scale. Raw photomask images measure 172,000×249,000 pixels, with a precision of 1.75 pixels per micrometer. When divided into 128×128 sub-images, the entire dataset becomes a collection of approximately 2.3 million images, reflecting the immense scale and precision required in semiconductor lithography.

Data Splitting and Availability. Despite the large volume of total dataset, the number of defective samples remains disproportionately small amount compared to normal images. To create a balanced and practical benchmark, we curated a smaller Sampled MaskVision with a 7:3 ratio of normal to defective images for the training and test sets. However, we also release the full MaskVision dataset (with its original imbalance) for researchers who wish to explore different sampling strategies or other research objectives.

Overall, this dataset aims to serve as a rigorous industrial scale benchmark, enabling the development of methods that handle both small defects and large volume inspection demands which is essential requirement for real time anomaly detection in advanced semiconductor manufacturing.

A. Data Composition

Each cropped image in this dataset has a resolution of 128×128 pixels, capturing critical microscopic details. In total, the sampled dataset contains 15,712 images. Among these, 11,000 are training normal images, 3,300 are test normal images, and 1,472 are test defect images. Overall, approximately 70% of the dataset is allocated to training, while the remaining 30% serves as the test set. This split ensures the model learns only from normal patterns and is evaluated on its ability to identify true anomalies.

B. Single Category Structure

Despite the variety of photomask designs, we organize all images under a single category:

- **Mixed Patterns**: Many images contain overlapping photomask patterns, making it difficult to assign a single pattern label or category without losing relevant context.
- **Consistency and Practicality**: A unified category reduces overhead in data preprocessing and pipeline complexity. In real world settings, product lines often evolve rapidly, so maintaining multiple pattern based categories becomes impractical.

Dataset	Train images / category	Test images / category	Categories
MVTec-AD [19]	242	115	15
VisA [24]	722	180	12
Kolektor SDD [19]	347	52	1
Sampled MaskVision (Ours)	11,000	4,712	1

TABLE I

COMPARISON OF SEVERAL ANOMALY DETECTION DATASETS, INCLUDING OUR NEWLY RELEASED MASKVISION DATASET SAMPLE. TEST IMAGES ARE INCLUDING BOTH NORMAL AND DEFECT IMAGES.

- **Complex Boundary Cases**: Anomalies may appear at the juncture of different patterns. Multiple categories could fragment the data and obscure defect patterns spanning category boundaries.

C. Challenges and Relevance

While each image is relatively small, the complexity of localizing multiple small defects makes this dataset particularly challenging:

- **Localization Complexity**: Photomask images often contain numerous closely spaced defects that many anomaly detection models merge into a single region. However, for photomask applications, each narrow defect area should ideally be identified separately rather than being grouped together. As indicated by the relatively modest AUPRO scores in table II, precisely segmenting each defect remains a difficult task.
- **Scale and Volume**: Even though individual images are only 128×128 pixels, industrial inspection might involve millions of these patches. Balancing both speed and accuracy is therefore critical for real world viability.

Overall, the MaskVision dataset serves as an industrial scale benchmark for anomaly detection, testing the ability of algorithms to identify weak defects under realistic manufacturing conditions, all while maintaining computational efficiency.

V. EXPERIMENTS

A. Experimental Setup

All experiments are conducted on a machine equipped with an Intel Core i9-13900KS CPU, 128GB of RAM, and an NVIDIA GeForce RTX 4090 GPU. Each compared model was trained and tested on this setup to ensure consistent and fair benchmarking of computational efficiency.

Datasets: We evaluated our approach on two primary datasets, including:

- *MVTec-AD* [4]: Consists of 15 categories with 3,629 training images (all normal) and 1,725 test images (both normal and anomalous). Ground truth masks are available for segmentation evaluation.
- *MaskVision Dataset (Section IV)*: A specialized industrial dataset with normal and defective images of photomask patterns. Unlike MVTec-AD, no data augmentation was applied to this dataset.

Augmentation: For the MVTec-AD datasets, we used rotation, gamma correction, Gaussian noise, and ISO noise to increase the variability of the training samples. By contrast, we used the original images for the MaskVision dataset to preserve its intrinsic characteristics.

B. Speed and Computational Efficiency

In many industrial applications, efficiency in anomaly detection is just as vital as accuracy, especially when inspecting large volumes of data in real time. To assess the computational efficiency and scalability of various methods, we conducted experiments on the MVTec-AD [4] by augmenting

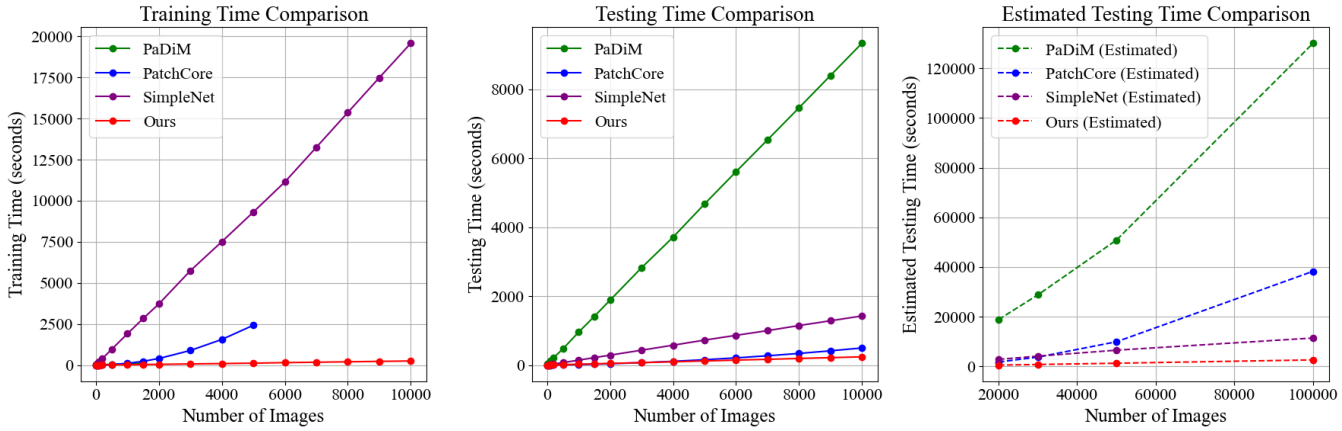

Fig. 3. A comparative analysis of training and testing times for different anomaly detection methods as a function of the number of images using augmented MVTec-AD [4] dataset. The left graph shows that our method trains drastically faster than other models, including deep learning–based approaches such as SimpleNet. In the center graph, our testing time is slightly slower than PatchCore [16] for smaller datasets, but due to its lower computational complexity, our approach becomes faster as the number of images increases. The right graph(log scale) provides an estimated testing time based on center graph's data for even larger datasets, illustrating that our method is better suited for large scale industrial scenarios where efficiency is foremost.

the number of images to over 10,000. As shown in fig. 3, these experiments highlight how different anomaly detection approaches handle increasing dataset sizes.

Our comparison specifically targets feature embedding based methods, since reconstruction based methods generally incur higher computational costs during both training and testing. Of particular note, PaDiM could only be trained on up to 1,500 augmented images before encountering memory overflows, and PatchCore reached its limit at 5,000 images. Beyond these points, the methods could not proceed with training due to insufficient resources, underscoring their limited scalability on industrial scale data.

In contrast, our proposed method does not require training a deep neural network, leading to a nominal linear growth in testing time and stable memory usage, even as the dataset size reaches 10,000 images. While certain approaches sometimes achieve marginally higher accuracy, their exponential or polynomial scaling in computational requirements can make them impractical for real world, large scale scenarios. The ability of our method to balance high performance with efficient resource utilization makes it a more feasible choice for industrial environments, where datasets can grow to millions of images.

Method	Pixel AUPRO
SimpleNet [13]	0.852
PaDiM(*) [7]	0.872
PatchCore(*) [16]	**0.901**
Ours	<u>0.889</u>

TABLE II

RESULTS ON THE MASKVISION DATASET, REPORTING PIXEL LEVEL AUPRO SCORE. SCORES IN BOLD INDICATE THE HIGHEST PERFORMANCE, WHILE UNDERLINED SCORES REPRESENT THE SECOND BEST. METHODS MARKED WITH (*) COULD NOT BE TRAINED ON THE MACHINE DESCRIBED IN SECTION V-A DUE TO MEMORY OVERFLOWS AND WERE INSTEAD TRAINED ON A WORKSTATION EQUIPPED WITH AN A100 GPU.

C. Evaluation on MaskVision Dataset

We tested our approach on the MaskVision dataset, which is uniquely challenging due to its high resolution images and dense photomask patterns. The primary difficulty lies in precisely localizing multiple small defects, each occupying a narrow region. As illustrated in fig. 1, the bottom row shows the VBAD model's predicted defect area (red) overlapped on the ground truth mask (blue). While the ground truth regions are extremely confined, the model's predictions tend to group nearby defects into a single broader region. This phenomenon occurs with other anomaly detection models as well. In practical photomask inspection, however, it is essential to separate each defect individually rather than merging them into one large segment.

Looking at table II, we see that the best pixel level AUPRO achieved by PatchCore [16] is 0.901, while VBAD attains 0.889. Although these scores are indicative of reasonably accurate localization, the ultimate goal is to detect each defect area with fine granularity. Achieving such defect segmentation is especially crucial in semiconductor manufacturing, where even a minuscule oversight can lead to costly yield losses.

Overall, PatchCore [16] leads in detection metrics but has substantial memory requirements, which can be challenging in high-volume industrial scenarios. PatchCore could not be run on our machine in which is described in Section V-A due to its high memory requirements. By contrast, our method offers competitive accuracy while using far fewer computational resources, making it well suited for real time deployment where efficiency is key.

D. Evaluation on MVTec Datasets

Table III compares the average image level AUROC scores on MVTec-AD for various anomaly detection methods. Among the listed approaches, GLASS achieves the highest

Method	Image AUROC	Pixel AUROC
AE-SSIM [3]	0.869	0.694
RIAD [26]	0.916	0.942
DRAEM [25]	0.98	0.973
CutPaste [12]	0.962	0.960
SimpleNet [13]	0.996	0.981
PaDiM [7]	0.969	0.975
PatchCore [16]	0.991	0.981
GLASS [10]	**0.999**	**0.993**
Ours	0.968	0.970

TABLE III

COMPARISON OF AVERAGE IMAGE AND PIXEL LEVEL AUROC ON MVTEC-AD, COMPUTED BY AGGREGATING SCORES ACROSS ALL CATEGORIES IN EACH DATASET.

average performance on MVTec-AD (99.9%), while PatchCore also demonstrates strong results (99.1% on MVTec-AD). Our method gets 96.8% on MVTec-AD, placing it within a competitive, but not reaching the top scores.

Notably, our method operates in DNN training free manner, in contrast to certain reconstruction based and embedding based methods that require DNN training or fine-tuning. This design choice yields significant computational advantages, especially on large datasets, at the cost of a marginal decrease in accuracy.

VI. CONCLUSIONS

We presented Vector Binning Anomaly Detection (VBAD), a novel and scalable anomaly detection method that leverages pretrained features and transition matrices for low computational overhead. We also propose a new anomaly detection dataset, MaskVision, that features challenging semiconductor inspection scenarios with repetitive patterns, subtle defects, and large-scale deployment potential. Experiments reveal both the strengths and the limitations of other models, especially regarding memory and computational cost on large data. By contrast, VBAD scales more effectively than the compared methods, albeit with a minor trade-off in accuracy.

REFERENCES

[1] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. Image anomaly detection and localization with position and neighborhood information. arXiv preprint arXiv:2211.12634, 2(5):6, 2022.

[2] Christoph Baur, Stefan Denner, Benedikt Wiestler, Nassir Navab, and Shadi Albarqouni. Autoencoders for unsupervised anomaly segmentation in brain mr images: a comparative study. Medical Image Analysis, 69:101952, 2021.

[3] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. arXiv preprint arXiv:1807.02011, 2018.

[4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9592–9600, 2019

[5] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. International Journal of Computer Vision, 130 (4):947–969, 2022.

[6] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint arXiv:2005.02357, 2020.

[7] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In International Conference on Pattern Recognition, pages 475–489. Springer, 2021.

[8] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 98–107, 2022.

[9] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pages 665–674, 2017.

[10] Qiyu Chen, Huiyuan Luo, Chengkan Lv, and Zhengtao Zhang. A unified anomaly synthesis strategy with gradient ascent for industrial anomaly detection and localization. In European Conference on Computer Vision, pages 37–54. Springer, 2024.

[11] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. IEEE Access, 10: 78446–78454, 2022.

[12] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9664–9674, 2021.

[13] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 20402–20411, 2023.

[14] Arian Mousakhan, Thomas Brox, and Jawad Tayyub. Anomaly detection with conditioned denoising diffusion models. arXiv preprint arXiv:2305.15956, 2023.

[15] Eduardo HP Pooch, Pedro Ballester, and Rodrigo C Barros. Can we trust deep learning based diagnosis? the impact of domain shift in chest radiograph classification. In Thoracic Image Analysis: Second International Workshop, TIA 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings 2, pages 74–83. Springer, 2020.

[16] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 14318–14328, 2022.

[17] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In International conference on machine learning, pages 4393–4402. PMLR, 2018.

[18] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. Advances in neural information processing systems, 12, 1999.

[19] Domen Tabernik, Samo Šela, Jure Skvarč, and Danijel Skoˇcaj. Segmentation-Based Deep-Learning Approach for Surface-Defect Detection. Journal of Intelligent Manufacturing, 2019.

[20] Hui Zhang, Zheng Wang, Zuxuan Wu, and Yu-Gang Jiang. Diffusionad: Norm-guided one-step denoising diffusion for anomaly detection. arXiv preprint arXiv:2303.08730, 2023.

[21] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In International conference on learning representations, 2018.

[22] Peng Xing and Zechao Li. Visual anomaly detection via partition memory bank module and error estimation. IEEE Transactions on Circuits and Systems for Video Technology, 33(8):3596–3607, 2023.

[23] Ke Yan, Jinzheng Cai, Adam P Harrison, Dakai Jin, Jing Xiao, and Le Lu. Universal lesion detection by learning from multiple heterogeneously labeled datasets. arXiv preprint arXiv:2005.13753, 2020.

[24] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pretraining for anomaly detection and segmentation. In European Conference on Computer Vision, pages 392–408.Springer, 2022.

[25] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In Proceedings of the IEEE/CVF international conference on computer vision, pages 8330–8339, 2021.

[26] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Reconstruction by inpainting for visual anomaly detection. Pattern Recognition, 112:107706, 2021.