OneN: Guided Attention for Natively-Explainable Anomaly Detection

Pasquale Coscia*, Angelo Genovese, Vincenzo Piuri, Fabio Scotti

Department of Computer Science, Università degli Studi di Milano, 20133, Milan, Italy

Abstract

In industrial computer vision applications, anomaly detection (AD) is a critical task for ensuring product quality and system reliability. However, many existing AD systems follow a modular design that decouples classification from detection and localization tasks. Although this separation simplifies model development, it often limits generalizability and reduces practical effectiveness in real-world scenarios. Deep neural networks offer strong potential for unified solutions. Nonetheless, most current approaches still treat detection, localization and classification as separate components, hindering the development of more integrated and efficient AD pipelines. To bridge this gap, we propose OneN (One Network), a unified architecture that performs detection, localization, and classification within a single framework. Our approach distills knowledge from a high-capacity convolutional neural network (CNN) into an attention-based architecture trained under varying levels of supervision. The resulting attention maps act as interpretable pseudo-segmentation masks, enabling accurate localization of anomalous regions. To further enhance localization quality, we introduce a progressive focal loss that guides attention maps at each layer to focus on critical features. We validate our method through extensive experiments on both standardized and custom-defined industrial benchmarks. Even under weak supervision, it improves performance, reduces annotation effort, and facilitates scalable deployment in industrial environments.

*Corresponding author.

Email addresses: pasquale.coscia@unimi.it (Pasquale Coscia), angelo.genovese.unimi.it (Angelo Genovese), vincenzo.piuri@unimi.it (Vincenzo Piuri), fabio.scotti@unimi.it (Fabio Scotti)

Keywords: Anomaly detection, Attention mechanism, Knowledge distillation, Generative model, Vision transformer

1. Introduction

Advanced manufacturing is increasingly adopting artificial intelligence (AI) to enhance production processes, ensure product quality, and support the maintenance of operational equipment [1]. Among its many applications, AI is being leveraged to predict machine faults (predictive maintenance) [2], detect anomalies in sensor data (anomaly detection) [3], and optimize production efficiency (process optimization) [4].

Vision-based automated defect detection systems [3] represent a key application area, targeting both *structural* anomalies (*e.g.*, physical damages) and *logical* anomalies (*e.g.*, omissions or misplacements). In this context, three core tasks are typically addressed: detection, localization and classification. Each serves a distinct purpose within automated inspection systems and is often implemented using specialized neural network architectures [5]. Detection focuses on establishing whether an image contains an anomaly, without specifying its exact location or type. This is typically framed as a binary classification problem (normal vs. anomalous), where global image-level features are extracted to make a decision. While classification networks can be adapted for detection, autoencoders (*e.g.*, U-Net [6]) are typically employed to generate anomaly heatmaps, which can then be thresholded to produce an image-level anomaly score. The key question in detection is *"Is this image anomalous?"*. Localization, instead, aims to identify the specific regions within an image that correspond to an anomaly, providing a pixel-wise defect segmentation. This task requires a network able to predict an anomaly mask to highlight the exact spatial extent of the defect. Localization is essential in applications where interpretability is crucial, such as industrial quality control [1]. The key question addressed by localization is *"Where is the anomaly in the image?"*. Finally, classification refers to assigning a label to an input image from a predefined set of categories, such as different types of defects in an industrial setting (*e.g.*, scratch, hole, or a generic damaged label). A classification network, such as a ResNet model [7], is often used to learn features that help discriminate normal from abnormal categories. In this case, the main question addressed is *"What type of anomaly is present?"*.

Conventional techniques [8] often lack robustness and accuracy, prompting the development of supervised [9, 10] and unsupervised [11, 12, 13] methods that offer substantial improvements. Multi-class unified anomaly detection (MUAD) approaches [14] further enhance scalability by generalizing across multiple classes instead of relying on separate models. This is achieved by capturing shared semantic patterns among categories [15, 16, 17]. However, they are typically designed to handle a single task at a time and often exhibit a significant performance degradation under distribution shifts. Consequently, multiple specialized models are frequently required, resulting in increased computational overhead and inefficiencies in deployment. For example, Hu *et al.* [5] employ a supervised framework composed of multiple distinct networks to address each task individually.

As a potential remedy, explainable AI (XAI) techniques [18] provide insights into neural network decisions and can facilitate both detection and localization within a unified framework. More specifically, visual explanation methods analyze internal network representations to identify image regions that are most influential for a given prediction. CAM-based techniques [19, 20, 21], for example, generate class activation maps, often guided by gradients, that highlight salient features contributing to a model's output. While these approaches were initially conceived for CNNs, attention-based mechanisms such as self-attention in Vision Transformers (ViTs) [22] have shown promise in capturing similarly informative patterns [23, 22, 24]. Despite their advantages, significant challenges persist in jointly enhancing explainability and anomaly detection performance. ViT-based models generate informative attention maps and achieve strong classification performance; however, their adoption in industrial settings remains limited for two key reasons: *(i)* self-attention mechanisms do not inherently localize anomalous regions, and *(ii)* their high computational requirements make them unsuitable for low-latency, energy-constrained environments [24]. To overcome these limitations, we propose a family of attention-based models tailored to diverse industrial requirements, along with a progressive guidance mechanism for self-attention that reduces dependence on heavy supervision, which is often necessary in prior methods [10, 5].

Knowledge distillation (KD) has been employed to improve the performance of compact models with limited representational capacity, offering solutions that balance accuracy with fast inference and low resource consumption, which are key priorities in industrial applications. [25, 26]. Traditional KD approaches [27] are typically designed for homogeneous architectures, such as CNN-to-CNN or Transformer-to-Transformer, where representation spaces are aligned. A common technique, logits-matching, is simple but may degrade student performance when teacher predictions are uncertain or noisy [28]. Distillation between heterogeneous architectures, like CNNs and ViTs, introduces additional challenges. CNNs gradually expand the receptive field through feature downsampling and spatial convolutions, whereas ViTs achieve a global receptive field immediately via self-attention. Their block structures also differ, CNNs are stage-specific, while ViTs use uniform blocks. Furthermore, CNNs typically use batch or group normalization, while ViTs rely on layer normalization, making feature or relation alignment more difficult. For these reasons, the application of KD is often limited to specific architectures and industrial tasks [29].

To address these challenges, this work proposes a unified framework that combines classification, detection, and localization into a single pipeline. Our approach employs KD to transfer knowledge from a high-capacity network to an attention-based architecture, enabling it to classify both normal and defective samples while using the ViT's attention maps to detect anomalies and generate segmentation masks. Given the strong performance of convolutional neural networks on small-scale datasets, we use a CNN as the teacher. We then distill knowledge into the student model using synthetic data generated by state-of-the-art AD generative models. To enhance localization performance when segmentation masks are available, we introduce a novel progressive masking mechanism during training. In this regard, we leverage the student's self-attention by including synthetic masks, either directly or by converting them into weakly annotated bounding boxes, which are easier to collect. By integrating classification, detection, and localization with weakly annotated mask supervision and adaptable architectures, the approach effectively addresses diverse industrial requirements. It also offers a practical trade-off between annotation cost and performance.

We summarize our main contributions as follows:

- We propose OneN, a unified framework based on KD that transfers knowledge from a convolutional teacher network to a ViT-based student, enabling a single model to jointly perform detection, localization and classification tasks.

- We introduce a progressive focal loss mechanism that guides both shallow and deeper attention layers to effectively localize structural anomalies.

- We investigate the use of synthetic images generated by diffusion-based models for anomaly detection, and define multiple evaluation protocols and input settings to reflect the diverse requirements of real-world industrial production lines. We compare our approach against multiple architectures currently used in such settings.

- We demonstrate that attention maps from a ViT model can be effectively used to detect anomalies without requiring additional networks or decoupling classification and detection/localization tasks. Our approach also generalizes well across various input configurations and industrial scenarios. Furthermore, we show that weak supervision (*e.g.*, bounding boxes) remains effective, significantly reducing annotation cost while maintaining competitive performance.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 presents our proposed model, based on KD and attention maps to detect anomalies using one architecture. Section 4 presents the main input settings and our results, along with a comprehensive analysis of the effectiveness of synthetic data for anomaly detection. Finally, Section 5 concludes the paper and discusses possible future research directions.

2. Related Work

Anomaly detection in industrial contexts aims to identify out-of-distribution (OOD) samples, wherein anomalous instances show statistical properties that deviate from the distribution of the available normal data [30, 31]. Given the relative ease of collecting normal samples, unsupervised methods have been developed to localize anomalies exclusively based on normal data distributions [32, 33]. Distribution-based approaches [34, 35] enhance feature alignment by leveraging pre-estimated statistical representations or memory banks. Patch-based techniques [36, 37], which perform fine-grained analyses at the patch level, have demonstrated improved sensitivity to subtle defects. Furthermore, recent advances have introduced multi-modal frameworks that integrate visual and linguistic modalities, offering enriched semantic understanding for anomaly detection [38, 39, 40]. Although these methods demonstrate strong performance in detecting anomalies, they continue to face significant challenges in meeting industrial requirements, particularly in terms of latency and efficiency [41, 42].
In the following, we provide a detailed overview of the main works related to our proposed approach.

2.1. Knowledge Distillation

Knowledge Distillation (KD) is widely employed to transfer learned knowledge from a large, well-performing teacher model to a compact student network. While logit-based KD is straightforward methodology to apply [27], feature-based distillation between heterogeneous architectures, especially from CNNs to Vision Transformers, remains challenging due to structural mismatches.

Touvron *et al.* [24] introduce a distillation token into the ViT architecture and train it using hard-label distillation, achieving performance on par with CNNs on mid-sized datasets while improving throughput. However, this approach benefits more from CNN-based teachers due to their inductive biases. Yang *et al.* [43] demonstrate that direct feature alignment between CNNs and ViTs degrades performance and propose a hybrid strategy: mimicking shallow features and employing a generative approach for deeper layers. TinyViT [44] defines a logit-based KD strategy for compressing large ViTs using sparse soft labels and constrained local search across embedding size, block depth, and channel count. Kang *et al.* [45] improve KD by aligning both the classification token and its attention maps to patches, although their method requires manual layer selection. Zheng *et al.* [46] address the CNN-to-ViT gap by using global average pooling and multi-head self-attention to render CNN features compatible with ViTs. Lin *et al.* [47] invert the direction, distilling from ViTs to CNNs via a teacher collaboration strategy that combines ViT- and CNN-based guidance using cross-attention. OFA-KD [28] aligns logits and early-exit branches across heterogeneous models, using adaptive weights to prevent learning from less confident teachers and increase entropy during training.

In the industrial context, specific methods have emerged to address unique domain challenges. TSKD [26] proposes a semantic-aware distillation at three levels: capturing intra- and inter-class variation through graph modeling, leveraging global semantics via attention maps and expert priors, and including cross-level responses among features, relations, and logits. Zhao *et al.* [48] build upon DeiT [24] to transfer spatial knowledge from CNNs to ViTs using dense predictions, avoiding direct feature matching. Zhang *et al.* [25] employ reverse distillation using one teacher and two students to capture local and global structures for unsupervised anomaly detection. A contextual affinity loss is defined to maintain spatial similarity, although using multiple decoders may limit applicability in resource-constrained settings. RD++ [29] introduces a multi-task reverse distillation scheme where simulated perturbations, such as simplex noise, act as pseudo-anomalies, although they risk introducing artifacts during training. Similarly, ROADS [14] integrates prompt and vision into a reverse distillation framework to handle distribution shifts. EfficientAD [49] performs anomaly detection at both the logical and structural levels, prioritizing computational efficiency through the use of a student–teacher framework and an autoencoder for global image analysis. A lightweight student network is trained to replicate the features extracted by a pre-trained teacher network using only normal samples. Anomalies are then detected when the student network fails to accurately reproduce these features. However, EfficientAD may encounter limitations when the predictions of the student and teacher networks diverge, potentially leading to an increase in false positives. Feng *et al.* [50] overcome the computational demands of multi-step diffusion models by distilling a multi-step teacher diffusion model into a single-step student generator.

2.2. Explainable Vision Transformers

Self-attention enables rich global context modeling, but this also introduces challenges for interpretability. While straightforward techniques (*e.g.*, Grad-CAM [20] or LayerCAM [51]) cannot be directly applied to Vision Transformers (ViTs) due to their fundamentally different mechanisms for producing visual explanations, several methods have been proposed to enhance the interpretability of ViTs, primarily via attention maps [52, 53] or learned masking strategies [54]. Yu *et al.* [55] propose a Siamese framework with normalization and decomposition modules to generate diverse and discriminative class-specific attributes. TransCAM [56] combines convolutional and transformer-based layers to improve class activation maps (CAMs), using attention weights atop the Conformer architecture [57] to refine spatial localization. Efficiency- and interpretability-focused variants such as LeViT [58] and MiniViT [59] include convolutional-like pooling or layer-sharing techniques. These designs require heavy modifications to the network, aiming to balance interpretability, diversity, and training stability.

To improve features representations in the industrial domain, Zhang and Liu [60] present a self-supervised framework for homogeneous attention-based networks, using multi-view augmentations to improve feature transfer. UniAS [16] bridges CNNs and Transformers by transforming convolutional features into patch-based representations for coarse-to-fine segmentation, helping to model fine-grained variability. AnomalyDINO [61] applies patch-based similarity using memory banks to spot anomalies; however, the quality of the memory bank can impact detection reliability. DRAEM [62] employs two networks, a reconstructive module to model normality and a discriminative one to identify defects, though its reliance on synthetic out-of-distribution samples may limit generalization. PromptAD [63] integrates CLIP [64] to extract language-informed visual features. Its multi-branch architecture enables zero-shot detection by combining anomalous and non-anomalous representations, though it still requires samples from related anomaly classes for optimal performance.

2.3. Generative Models

Due to the scarcity of anomalous samples, generative models are often leveraged to compensate for the limitations of training data-driven approaches. While GAN-based methods [65, 66, 67] struggle to effectively capture both intra- and inter-class variability, recent efforts have shifted toward diffusion-based techniques [68, 69]. For example, AnoDiff [5] employs a latent diffusion model pre-trained on large-scale datasets to enable few-shot anomaly synthesis. It introduces spatial anomaly embeddings to disentangle appearance from location and includes adaptive attention reweighting to enhance alignment with ground-truth masks. By contrast, AnoGen [10] learns an explicit anomaly distribution, which is embedded and used to condition a diffusion model alongside bounding box information. This design facilitates weakly supervised learning by generating realistic anomalous samples. Compared to earlier works that often required separate models for each anomaly type, SeaS [70] proposes a unified generative model. It is able to produce diverse anomalies by leveraging a U-Net architecture with separation-and-sharing fine-tuning. To address the challenge of generating novel anomaly types, AnomalyAny [71] introduces a framework based on Stable Diffusion [72], enabling the generation of anomalies conditioned on a single normal sample and corresponding text descriptions. This approach can synthesize high-quality, previously unseen anomalies without the need for extensive training data. However, directly applying Stable Diffusion for anomaly generation may result in unrealistic patterns if not properly guided.

3. Method

CNNs rely on local receptive fields, while ViTs use attention maps to identify the most important image regions for classification. This property makes ViTs particularly suitable for anomaly detection, where both classification

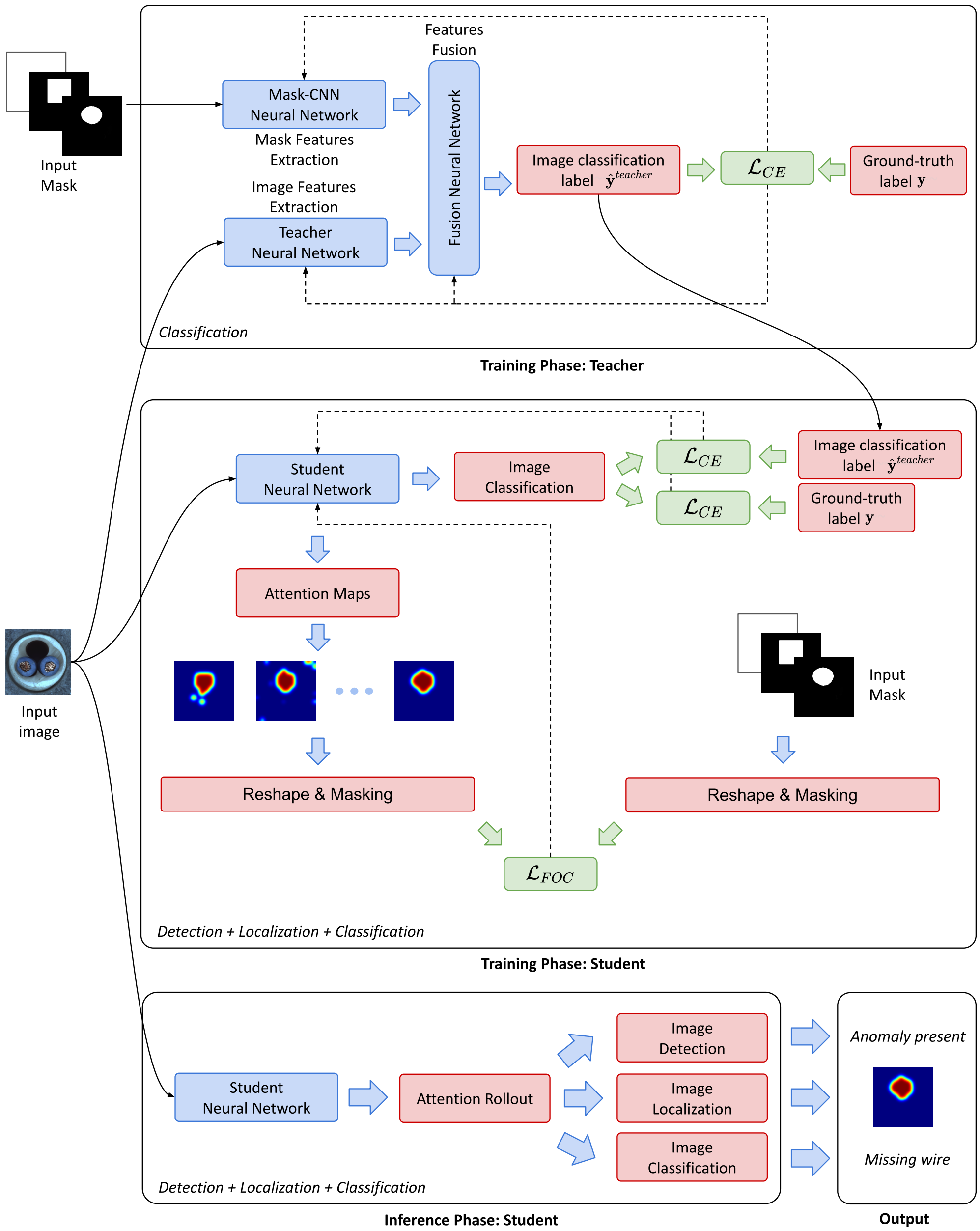


Figure 1: Training and inference pipelines of the proposed framework. The teacher is trained for classification only, while the student performs all tasks and is trained via knowledge distillation. Neural network components (in blue), functional modules (in red), loss functions (in green), and the backpropagation operation (dotted) are depicted. Best viewed in color.

and detection/localization of defective regions are required. For this purpose, we propose a unified framework that extends the capabilities of ViTs beyond classification. It leverages attention maps to highlight structural anomalies while employing a convolutional network-based teacher model to guide the training process through knowledge distillation. Due to the scarcity of anomalous samples in industrial scenarios, we consider diffusion-based generative models for synthesizing input image/mask pairs. In the following, we detail our approach (depicted in Fig. 1), describing both the teacher and student networks, and the progressive masking mechanism used for guided training.

Teacher Network. Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$, where (H, W) represents the image resolution and C the number of channels, we first train a CNN as a teacher model to classify the image as either normal or defective (more details about the different input settings are provided in Section 4). Let N_{def} denote the number of defect types within an object category T. To perform this classification task, we modify the final classification layer to predict $N_{def} + 1$ classes, discriminating between normal and defective instances. Given $\mathbf{y}$ and $\hat{\mathbf{y}}^{teacher}$ as the ground-truth and predicted labels, respectively, the teacher network is trained using a standard cross-entropy loss function, $\mathcal{L}_{CE}(\hat{\mathbf{y}}^{teacher}, \mathbf{y})$.

Additional supervision may be provided in the form of fine-grained annotations, such as pixel-level masks, or coarse-grained labels, such as bounding boxes. To leverage this information, we adopt a late fusion strategy, where RGB images and annotations are processed through separate branches before being concatenated at the classification layer. Specifically, the primary network branch extracts high-level representations from the image, while a dedicated branch, named Mask-CNN, processes the mask annotations, capturing spatial defect information. Further details about the mask branch are provided in Section 4. The extracted features are then concatenated and passed through a classification layer, allowing the model to integrate complementary information while separately processing each piece of information.

Student Network. Instead of employing additional explainability techniques, we adopt a ViT-based student network to inherently leverage its self-attention maps for defect localization and detection tasks. In the following, we provide a brief overview of its main elements essential to our work. More specifically, a ViT processes an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ by dividing it into smaller patches $\mathbf{I}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where (P, P) denotes the patch resolution. The number of resulting patches is given by $N = HW/P^2$. To perform classification, an additional learnable embedding (z_{class}) is prepend to the embedded patches. These patches are then processed by a transformer encoder, which applies normalization and multi-headed self-attention layers [73]. The input projections for the self-attention mechanism, namely key, query, and value, are computed as follows:

$$\mathbf{Q} = \mathbf{x}\mathbf{W}^K, \qquad \mathbf{K} = \mathbf{x}\mathbf{W}^Q, \qquad \mathbf{V} = \mathbf{x}\mathbf{W}^V, \tag{1}$$

where $\mathbf{W}^K$, $\mathbf{W}^Q$, and $\mathbf{W}^V$ are projection matrices in $\mathbb{R}^{d_{model} \times d_k}$. Typically, $d_k = d_{model}/h$, where $d_{model} = 512$ and h represents the number of attention heads (*e.g.*, $h = 8$). The attention function is computed as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} = \mathbf{A}\mathbf{V}, \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the attention weight matrix. The transformer consists of L stacked identical layers, each refining the previous representation. By extracting the attention maps corresponding to the classification token, one can visualize the regions influencing the classification decision. Since CNNs have been shown to be more efficient in low-data regimes, we rely on DeiT [24], which introduces a knowledge distillation mechanism tailored for ViTs, which includes an additional distillation token that interacts with other tokens in the self-attention mechanism and is explicitly trained using labels from a CNN-based teacher model. While classification results can be obtained from the final multi-layer perceptron (MLP), the image regions responsible for the prediction mainly rely on the attention maps $\mathbf{A}$. To interpret the regions influencing classification, we consider attention rollout [52]. Given that ViTs consist of multiple attention layers, attention rollout aggregates attention maps across layers by recursively multiplying them. This cumulative effect captures how much each input token contributes to the final prediction. More in details, for an input with $N + 2$ tokens (including the classification and distillation tokens) at the l^{th} layer, an attention map is defined as $\mathbf{A}^{(l)} \in \mathbb{R}^{(N+2)\times(N+2)}$. Attention rollout includes both direct attention and residual connections as $\tilde{\mathbf{A}}^{(l)} = \alpha_{roll}\mathbf{A}^{(l)} + (1 - \alpha_{roll})\mathbf{I}_{N+2}$. Finally, the aggregated attention map is computed recursively as $\mathbf{A}_{roll} = \tilde{\mathbf{A}}^{(1)} \times \tilde{\mathbf{A}}^{(2)} \times \cdots \times \tilde{\mathbf{A}}^{(L)}$, where N_a represents the total number of attention maps. The resulting matrix can be visualized as a heatmap, where each entry $\mathbf{A}_{roll}(i, j)$ represents the cumulative attention from token i to token j. In our work, we consider the image regions attended by the classification token, excluding the distillation patch, as it is only used in the knowledge transfer process.

Student losses. Our student network is trained using a combination of loss functions, addressing three distinct objectives: classification, localization/detection, and distillation. Specifically, both the classification and distillation objectives rely on the standard cross-entropy loss. We opt for a hard distillation approach, wherein the teacher's classification outputs serve as hard labels for the student, as follows:

$$\mathcal{L}_{\mathrm{KD}}^{\mathrm{student}} = (1 - \alpha_{KD})\mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{y}}^{student}, \mathbf{y}) + \alpha_{KD}\mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{y}}^{student}, \hat{\mathbf{y}}^{teacher}), \tag{3}$$

where $\hat{\mathbf{y}}^{student}$ represents the logits of the student networks associated with the classification head. Rather than employing a soft-distillation procedure, we choose hard-distillation for two main reasons: first, it has been shown to enable more effective knowledge transfer from a CNN-based teacher [24]; second, due to the limited number of classes for our experiments, typically less than 10 per object, a soft-distillation strategy would introduce additional noise that is not beneficial for our task.

Progressive Masked Focal Loss. Including additional supervision in the form of masks $\mathbf{M}$ can enhance the learning process by guiding attention maps toward the most relevant regions, such as defective areas. However, defective regions typically represent only a small fraction of an image, leading to a significant imbalance between defective and normal areas. To address this problem, we rely on the focal loss [74], following [5, 62], to improve the estimation of defective regions. This approach reduces the dominance of well-classified examples in the loss function, ensuring a greater focus on rare and difficult cases. The focal loss is defined as:

$$\mathcal{L}_{\mathrm{FOC}}(p_t) = -\alpha_t(1 - p_t)^{\gamma} \log(p_t), \tag{4}$$

where p_t denotes the model's estimated probability for the correct class, α_t is a balancing factor, and γ controls the attenuation of easy examples. To further enhance the robustness and generalization of attention maps, we introduce a progressive masked focal loss (PMFL) mechanism. While Yang *et al.* [43] propose a similar distillation procedure based on masking only specific layers via mimicking and generation procedures, our technique progressively applies a simpler adaptive masking across multiple attention layers while leveraging focal loss to improve optimization in imbalanced learning scenarios. In this way, we avoid explicitly defining shallow and deeper layers while providing a more robust generalization capability.

Our objective is to encourage attention maps, particularly in deeper layers, to focus on defective regions with higher precision. More specifically, for the i^{th} attention map $\mathbf{A}_i \in \mathbb{R}^{B \times P^2}$, where B is the batch size and P^2 is the total number of patches, we down-sample the target mask $\mathbf{M}$ to reduce computational overhead. A binary mask $\mathbf{M}_i \in \{0,1\}^{B \times P^2}$ is then sampled using a Bernoulli distribution with a head-dependent probability:

$$\mathbf{M}_i \sim \text{Bernoulli}(p_i), \quad p_i = p_{\min} + (p_{\max} - p_{\min}) \times \frac{i}{N_a}, \quad i = 1, ..., N_a, \quad (5)$$

where $p_{\min}, p_{\max} \in [0,1]$ define the minimum and maximum retention probabilities, respectively, and N_a represents the total number of attention maps. For example, with $p_{\min} = 0.1$ and $p_{\max} = 1.0$, only 10% of attention values are retained in the earlier layers, whereas 100% of the attention values are preserved in the deeper layers.

The masked out elements are removed from both the attention map and the target mask as follows:

$$\hat{\mathbf{A}}_i = \mathbf{A}_i \odot \mathbf{M}_i, \quad \hat{\mathbf{M}}_i = \mathbf{M} \odot \mathbf{M}_i, \quad (6)$$

where $\odot$ denotes element-wise multiplication. The final loss is computed using the focal loss function (we omit for simplicity the input parameters):

$$\mathcal{L}_{\text{PMFL}} = \frac{1}{N_a} \sum_i \mathcal{L}_{\text{FOC}}(\hat{\mathbf{A}}_i, \hat{\mathbf{M}}_i), \quad i = 1, ..., N_a. \quad (7)$$

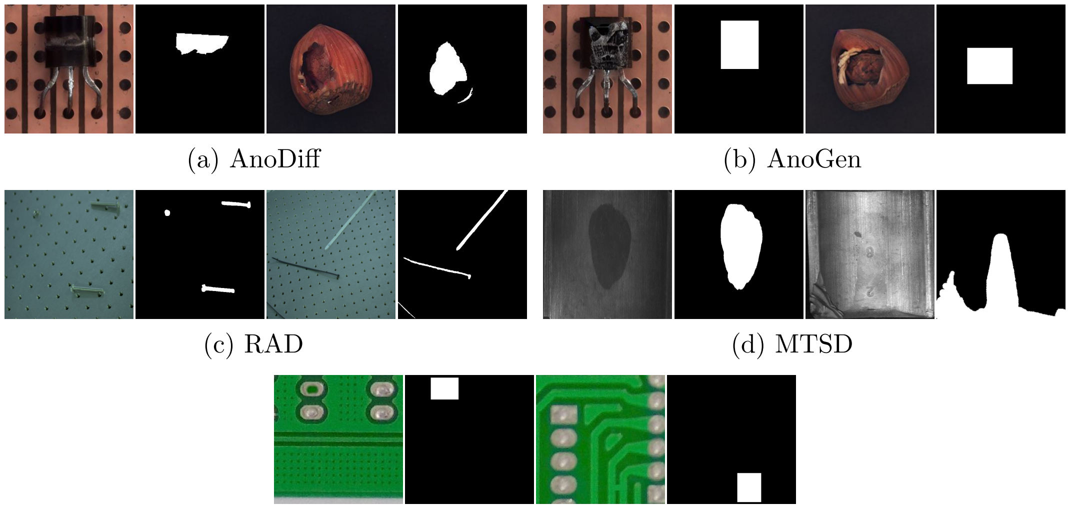
By progressively varying the masking ratio across layers, this approach ensures that deeper layers emphasize annotated defects while maintaining stable training. Finally, the overall loss function is given by:

$$\mathcal{L}^{\text{student}} = \mathcal{L}^{\text{student}}_{\text{KD}} + \beta \mathcal{L}^{\text{student}}_{\text{PMFL}}. \quad (8)$$

This formulation enables the student ViT model to effectively learn both classification and localization/detection tasks by leveraging teacher-guided distillation alongside attention-driven defect localization.

4. Results

Our analysis focuses on assessing the performance of the proposed approach across various input settings and comparing different architectures



(a) AnoDiff (b) AnoGen

(c) RAD (d) MTSD

(e) PCB

Figure 2: Defective samples from the selected datasets: (a) and (b) transistor (damaged case) and hazelnut (crack), (c) bolt and ribbon, (d) fray and break, and (e) missing hole and short.

for both classification and detection/localization tasks. Below, we present a comprehensive overview of the datasets used, the evaluation protocols applied, the experimental results obtained, and a series of ablation studies conducted to further examine the key components of our method.

Datasets. We consider four distinct datasets, collected under varying scenarios, to evaluate our framework and ensure its applicability across a range of real-world conditions. Firstly, we use the MVTec Anomaly Detection (MVTec AD) [75] dataset, a standard benchmark for evaluating industrial inspection methods. It consists of high-resolution images across 15 object and texture categories, including both normal and defective samples. Each category includes various types of anomalies, such as scratches, deformations, and contaminations, along with pixel-wise ground truth masks to enable precise defect localization. For training our models, we rely on synthetic images generated by two diffusion-based models specifically designed to solve both classification and detection/localization tasks, *i.e.*, AnoDiff [5] and AnoGen[1] [10]. Since synthetic datasets are usually preprocessed through

[1]As its cleaned version does not provide the *flip* defect for the metal nut category, we exclude this defective class from our experiments for this dataset.

a cleaning procedure, we use data from official repositories, including ~ 500 images/defect. As normal samples, we use the images provided by the MVTec AD dataset (~ 200 images/category). A comparison of the training images is shown in Figure 2a and Figure 2b. As test set, we consider the image/mask pairs from the official split.

Then, we consider the RAD (Robust Anomaly Detection) [76] dataset, which is specifically designed to evaluate the robustness of image anomaly detection methods under realistic industrial conditions. Unlike the MVTec AD dataset, which primarily focuses on detecting product defects, RAD emphasizes the identification of foreign objects on industrial work platforms. The dataset introduces several sources of imaging noise, *e.g.*, varying viewpoints (camera angles and object positions), uneven illumination (including bright spots and shadows), and image blur (caused by defocus and motion), which are often combined within the same image, making RAD a challenging benchmark for anomaly detection. The dataset comprises 286 normal samples, of which 213 are used for training and 73 for testing. In addition, it includes 1,224 abnormal samples across four categories: 327 for bolt, 293 for ribbon, 281 for sponge, and 323 for tape. In our study, all normal samples are combined to form a single good class, resulting in a total of five classes (*i.e.*, 1 normal + 4 abnormal). For each anomalous class, we adopt an approximate 75%/25% split for training and testing, respectively. Figure 2c shows two samples from this dataset.

Additionally, we test our framework against the Magnetic Tile Surface Defect (MTSD) [77] dataset, which is designed to detect surface defects in magnetic tiles under real-world industrial conditions. This dataset emphasizes low-contrast surface defects that occur in textured industrial materials, presenting unique challenges due to the high intra-class variance, irregular defect shapes, and complex illumination conditions. The dataset comprises 1,344 images, from which regions of interest (ROIs) are cropped to focus on five specific defect types: blowhole, crack, fray, break and uneven. The dataset also contains a defect-free (free) class. Pixel-level ground truth annotations are provided for each ROI, enabling precise saliency-based evaluation. In our study, we consider six classes (*i.e.*, 1 normal and 5 abnormal). For each class, we adopt an approximate 75%/25% split for training and testing, respectively. Figure 2d shows some images and masks from this dataset.

Finally, we consider the PCB defect [78] dataset, which contains high-resolution images of printed circuit board (PCB) defects and is specifically constructed for the task of tiny defect detection. It comprises 693 PCB

Table 1: Overview of input settings used in our experiments.

Input Setting	**Acronym**	**Image**		
		Defective	**Good**	**Background**
Standard	*STD*	✓	✗	✗
Standard + Good	*STD+G*	✓	✓	✗
Standard + Background	*STD+BG*	✓	✗	✓
Standard + Good + Background	*STD+G+BG*	✓	✓	✓
Binary	*B*	✓(Combined)	✓	✗
Binary + Background	*B+BG*	✓(Combined)	✓	✓

images with an average resolution of 2777×2138 pixels, each annotated with one or more instances of six common defect types: missing hole, mouse bite, open circuit, short, spur, and spurious copper. Due to the limited number of original samples and inherent class imbalance, extensive data augmentation was employed to enhance generalization and mitigate overfitting. As a result, the augmented dataset contains 10,668 images, partitioned into 8,534 training and validation samples, and 2,134 testing samples. This dataset presents significant challenges for defect detection due to the small size of the defects relative to the overall image dimensions and the presence of multiple defect types within single images. To address these challenges, we divide each 600×600 augmented image into nine non-overlapping patches, each treated as an individual sample. To reduce training time, we further sub-sample both the training and testing sets by randomly retaining only one-tenth of the patches. Figure 2e shows two samples of cropped PCBs from this dataset.

Classification Task. As reported in Table 1, we consider six different configurations, *i.e.*, variations of the input setup, to assess the effectiveness of the classification architectures. More specifically, STD refers to a standard configuration where only defective classes are considered. The number of defective classes for each category in the MVTec AD dataset is reported in Table 2. We modify this setup by also including normal samples ($STD{+}G$), allowing for a more comprehensive evaluation of the classifier's ability to discriminate between multiple classes. Since low-cost sensors may partially capture the object along with the background, we investigate the potential impact of background elements on classifier performance by including background images in the $STD{+}BG$ and $STD{+}G{+}BG$ configurations (a detailed description of the background class is provided below). Finally, we construct a binary scenario by grouping all defect types into a single class along with the normal samples (B), as well as an extended version that also includes

Table 2: Number of classes for each category in the MVTec AD dataset used in the *STD* input setting.

Objects	
Category	**Num. Classes**
Toothbrush	1
Bottle	3
Hazelnut, Metal Nut, Transistor	4
Capsule, Screw	5
Pill, Zipper	7
Cable	8
Textures	
Category	**Num. Classes**
Carpet, Grid, Leather, Tile, Wood	5

the background (*B+BG*). We report the standard top-1 accuracy, defined as the proportion of correctly classified test samples out of the total number of test samples. This metric is applied to both multi-class and binary classification tasks. The number of classes used in our classification experiments varies by MVTecAD category under different input settings. Under the *STD* setting, each defect type is treated as a separate class. The *STD+G* and *STD+BG* settings each introduce one additional class by adding normal or background samples, respectively, while the *STD+G+BG* setting includes both. In contrast, the *B* and *B+BG* settings use 2 and 3 classes, respectively, by merging all defect types into a single defective class and optionally including background images. The toothbrush category of the MVTec AD dataset is excluded from the *STD* setting due to having only a single defect type, while texture categories are not evaluated in settings involving background images. To comprehensively evaluate model performance, each configuration is assessed using two distinct protocols:

- **Training-aware (TR-AW) Strategy**: Model performance is tracked on the training set using the loss function, without access to validation or test data.

- **Test-aware (T-AW) Strategy**: Model performance is evaluated directly on the test set, without considering validation dynamics. While this may lead to overly optimistic results, it provides an estimate of the model's upper-bound performance. This strategy is adopted in [5, 62].

Background Class. To generate background-only images, we process a subset of randomly selected training (non-defective) images by applying binary masks that remove object regions while retaining peripheral areas. Since anomalies in the MVTec AD dataset typically dominate the central portion of the image, we preserve the left and right borders, which are more likely to contain background content. We then apply an inpainting algorithm [79] to fill the masked regions using surrounding pixels, thereby eliminating object-specific information and producing visually coherent background textures. This approach aims to preserve only background characteristics while minimizing the presence of residual object features. Although the RAD, MTSD, and PCB datasets do not explicitly contain objects, we decide to include this fictitious class to represent anomalies in the capturing process, *e.g.*, failures in cameras used to collect the data. In these cases, we adopt the same technique of MVTec AD dataset by only preserving left and right borders. Representative examples of the resulting background images are shown in Figure 3, featuring normal samples (top row) and background images (bottom row) generated by our simple image processing pipeline for three categories, hazelnut, cable, and transistor, from the AnoDiff dataset, and normal samples from the other datasets. Categories such as transistor and cable exhibit more complex backgrounds, making it challenging to extract them solely from image borders. For the MVTec AD dataset, the background class is defined only for object categories, as discriminating between object and background in texture categories is inherently ambiguous. Therefore, when background images are included in the input setting, texture categories are excluded from the evaluation, and results are reported only for the object categories.

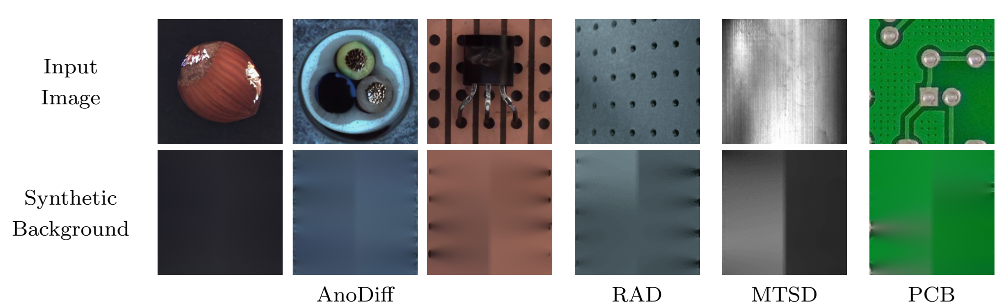


Figure 3: Normal and background samples generated for our datasets using our image processing pipeline.

Detection and Localization Tasks. Detection and localization tasks consist in identifying defective regions at both image and pixel levels. For detection and localization tasks, we employ metrics that better reflect spatial and score-based prediction quality. Specifically, we evaluate anomaly detection performance using three standard metrics: Area Under the Receiver Operating Characteristic Curve (AUC), Average Precision (AP), and Maximum F_1-max ($F1_{max}$). These metrics are computed at two levels of granularity, pixel-level and image-level. To assess spatial localization accuracy (pixel-level), we evaluate predicted anomaly maps against binary ground-truth masks. Both maps are flattened across the dataset to form pixel-wise score-label pairs. Based on these, we compute each metric as follows:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(t)\, d\mathrm{FPR}(t), \tag{9}$$

$$\mathrm{AP} = \int_0^1 \mathrm{Precision}(\mathrm{Recall})\, d(\mathrm{Recall}), \tag{10}$$

$$\mathrm{F}_1(t) = \frac{2 \cdot \mathrm{Precision}(t) \cdot \mathrm{Recall}(t)}{\mathrm{Precision}(t) + \mathrm{Recall}(t)}, \tag{11}$$

$$\mathrm{F}_1\text{-max} = \max_{t \in [0,1]} \mathrm{F}_1(t). \tag{12}$$

In the above equations, TPR and FPR denote the true positive and false positive rates. To evaluate anomaly detection at the image level, each image is assigned a scalar anomaly score defined as the maximum value over its predicted anomaly map. Given binary image-level labels $\mathbf{y} \in \{0, 1\}$ indicating normal ($\mathbf{y} = 0$) or anomalous ($\mathbf{y} = 1$) samples, we compute the same set of metrics (AUC, AP, and F_1-max) based on the predicted scores and corresponding ground-truth labels.

Since detection and localization tasks are usually performed using both defects and normal samples, we evaluate these tasks only using one input setting, *i.e.*, *STD+G*. While the performance of ViT-based architectures is evaluated using attention rollout [52], CNN-based architectures are assessed using CAM-based methods. Specifically, we employ Grad-CAM [20], Grad-CAM++ [80], LayerCAM [51], and Score-CAM [21], applied to the final convolutional layer of each architecture.

Baselines. The teacher model is based on the RegNet-Y-16GF architecture [81], pretrained on ImageNet-1k, and is used to extract features from

Table 3: Model specifications and performance of the architectures investigated in our study.

Model Complexity Comparison					
Category	**Model**	**Params** (M)	**FLOPs** (G)	**Latency** (ms)	**Throughput** (images/s)
Lightweight	SqueezeNet 1.0	0.74	1.48	10.94	91.60
	MobileNet V3	4.21	0.44	14.74	68.55
ResNet	ResNet-34	21.29	7.34	19.79	51.10
	ResNet-152	58.15	23.09	67.08	14.99
ViT	ViT-B-16 (S)	22.01	8.96	18.66	53.71
	ViT-B-16 (B)	57.45	23.51	50.80	19.75
	ViT-B-16 (L)	85.80	35.15	75.00	13.36
	CSKD-Ti	5.53	2.17	11.84	86.39
	CSKD-S	21.67	8.54	25.64	39.44
	CSKD-B	85.81	33.89	76.11	13.20
	OneN-S	22.02	8.65	18.74	53.43
	OneN-B	57.46	22.67	50.37	19.89
	OneN-L	85.81	33.89	76.05	13.17

RGB input images, while a DeiT-B [24] network is used as student. Our approach is compared against models with similar computational complexity, including ResNet-34, ResNet-152, and ViT-B-16. For the ViT-B-16 architecture, we consider three variants by limiting the number of transformer layers to 3 (S), 8 (B), and 12 (L). Additionally, we compare our models with CSKD [48], using its tiny (Ti), small (S), and base (B) variants. This approach is closely aligned with ours, as CSKD performs spatial-wise knowledge transfer from a CNN to a ViT. Similarly, we propose three variants of our architecture, *viz.*, small (OneN-S), base (OneN-B) and large (OneN-L), each containing 3, 8 and 12 transformer encoder layers, respectively. This design choice is intended to support a range of deployment scenarios with varying computational constraints. For example, OneN-L is tailored for high-accuracy applications, such as off-line inspections and detailed defect analyses, where computational resources are less restricted. In contrast, OneN-S is optimized for real-time operations in resource-constrained environments, including live production line monitoring and embedded industrial systems. Positioned between these extremes, OneN-B provides a balanced trade-off between accuracy and efficiency, making it suitable for semi-real-time inspection tasks or systems with moderate hardware capabilities. Table 3 presents the computational complexity of each architecture in terms of the number of parameters, FLOPs (Floating-Point Operations), latency, and throughput, all measured on an Intel® Core™ i7-7800X @ 3.50 GHz CPU. Among the attention-based networks, CSKD-Ti is the smallest and most parameter-efficient model, while the B and L variants are the most resource-intensive. ResNet-34 is comparable to our OneN-S model, while ResNet-152 aligns more closely with our medium-sized OneN-B. We also include two parameter-efficient models, MobileNet V3 and SqueezeNet 1.0, which are commonly used in resource-constrained environments. All models are pre-trained on the ImageNet-1k dataset. Our proposed models remain comparable to their ViT-based counterparts, differing only by the inclusion of an additional distillation token while the parameter-efficient models offer the lowest complexity.

Implementation Details. We resize the input images to 224×224 pixels, use the Adam optimizer [82] with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 0.0004, and a batch size of 128. Weight decay is set to 0.05, and early stopping is applied with a patience of 35 epochs. For data augmentation, we apply random cropping (ensuring that the defect area is always included when the image contains a defect), Gaussian blurring, horizontal flipping, and random rotation up to 20 degrees, each with a 50% probability. The focal loss is used with parameters $\alpha_t = 1$ and $\gamma = 2$ while we train the student network setting α_{KD} to 0.5.

Following previous studies [62, 5, 25], we employ a one-model-per-category setting, training a separate model for each category. Thus, for each experiment, we train 15 models and report the mean values of the evaluated metrics. We omit standard deviation for brevity, as our primary focus is on overall trends and relative performance across methods. To improve detection and localization performance, we compute the final attention map using the attention rollout method [52], applying a maximum operation and discarding the lowest 75% of attention values. We also set α_{roll} to 0.5. Furthermore, the Mask-CNN processes the input mask through a dedicated branch consisting of three convolutional layers with increasing channel sizes ($32 \rightarrow 64 \rightarrow 128$), each followed by Batch Normalization, ReLU activation, and MaxPooling for downsampling. An adaptive average pooling layer then generates a 128-dimensional feature representation, which is concatenated with features extracted from the RGB image and passed through a fully connected layer for final classification.

Quantitative Results. In the following, we first present our quantitative results by reporting the performance of two reference classifiers for

Table 4: Classification performance for two simple classifiers across different input settings used in our experiments using the MVTec AD dataset.

Input Setting	**Random Classifier** (%)	**Mean CV**	**Majority Classifier** (%)
STD	20.67	0.09	23.75
STD+G	19.19	0.34	31.43
STD+BG	20.44	0.07	31.31
STD+G+BG	16.41	0.36	33.99
B	50.00	0.47	73.31
B+BG	33.33	0.46	72.86

baseline comparison. Then, for each dataset considered, we describe both the classification and detection/localization results.

Table 4 presents the performance of two baseline classifiers: a random classifier and a majority-class classifier on the MVTec AD dataset. The random classifier assigns labels uniformly at random across all classes, while the majority-class classifier always predicts the most frequent class observed in the test set. To contextualize the impact of class imbalance, we also report the coefficient of variation (CV) of the class distribution, which quantifies the relative dispersion of class frequencies (with CV = 0 indicating perfectly balanced classes and higher values reflecting greater imbalance). We recall that $0 \leq \mathrm{CV} \leq \sqrt{n-1}$, where n is the number of classes. On average, the random classifier achieves a classification accuracy of approximately 20% across input settings involving more than three classes. The reported CV values indicate that the *STD* and *STD+BG* settings have relatively balanced class distributions, while the remaining settings show greater imbalance. In some cases, a single class dominates the test set, accounting for as much as 30% to 70% of all instances.

Table 5 presents the classification accuracies for the AnoGen dataset under the TR-AW and T-AW evaluation protocols across various input settings. This dataset includes weakly annotated masks that enhance classification performance across most settings. Our OneN-L model achieves the best performance overall, consistently ranking first or second among the evaluated methods. We observe that for input settings with only 2 or 3 classes, OneN-S, the smallest model in our family, achieves the highest performance under the TR-AW evaluation protocol. In contrast, the CSKD models and MobileNetV3 architectures perform close to random, indicating limited learning. ViT models also generally perform poorly under this protocol, suggest-

Table 5: Average classification performance on the AnoGen dataset under TR-AW and T-AW strategies. OneN results are reported with (☐) and without (✗) weak masks. Best and second-best scores per column are bolded and underlined, respectively.

		Accuracy (%)					
Model	**Mask**	***STD***	***STD+G***	***STD+G+BG***	***STD+BG***	***B***	***B+BG***
MobileNet V3	✗	20.65/32.92	19.26/37.34	19.59/35.96	17.65/35.49	33.88/58.63	19.46/43.64
SqueezeNet 1.0	✗	34.88/49.13	42.26/53.09	32.97/46.97	37.71/51.19	40.72/75.34	40.56/78.27
ResNet-34	✗	39.47/53.23	37.38/51.51	39.12/48.39	40.98/55.68	44.35/79.13	41.16/78.45
ResNet-152	✗	37.92/53.86	27.81/50.97	33.75/47.39	27.69/53.99	38.02/73.48	28.41/73.82
ViT-B-16 (S)	✗	42.57/48.87	40.94/48.54	42.25/43.06	40.91/51.53	28.31/69.21	29.86/71.86
ViT-B-16 (B)	✗	47.46/51.30	43.49/51.07	41.75/44.70	47.88/50.26	28.59/74.59	32.11/68.91
ViT-B-16 (L)	✗	48.98/53.01	40.90/51.23	41.46/47.01	46.32/52.47	29.97/74.09	30.62/74.83
CSKD-Ti	✗	20.82/23.20	19.41/21.28	18.24/22.55	29.30/34.07	45.18/67.82	51.37/74.61
CSKD-S	✗	20.82/23.20	19.41/21.28	18.24/22.55	29.30/34.07	45.18/67.82	51.37/74.61
CSKD-B	✗	22.83/26.35	20.78/26.13	22.17/22.77	27.29/33.76	57.23/65.06	36.91/74.62
OneN-S	✗	32.50/37.31	28.38/34.55	30.31/29.03	27.82/46.41	**63.01**/75.03	**55.44**/74.94
OneN-B	✗	45.73/53.12	47.11/53.78	43.60/46.88	36.65/52.97	52.10/76.92	38.73/75.17
OneN-L	✗	47.87/56.49	45.95/55.12	**48.65**/52.39	38.38/**55.79**	51.27/77.97	33.44/76.14
OneN-S	☐	36.52/44.80	33.19/39.14	33.52/37.98	38.47/41.35	58.19/63.06	52.88/76.93
OneN-B	☐	45.76/57.27	44.35/54.29	37.91/43.55	46.75/50.31	62.66/77.70	53.54/78.49
OneN-L	☐	**49.32/58.04**	**51.31/61.03**	44.42/**54.41**	**50.59**/52.50	56.94/**83.24**	44.08/**80.96**

ing they learn incorrect patterns. This highlights a potential sensitivity of ViT architectures to the absence of strong supervision or contextual cues, or possibly a reliance on large-scale datasets for effective training. Under the *STD* input setting, both our OneN-L model and the ViT-B-16 (L) network show comparable performance under the TR-AW protocol. OneN-S, when equipped with masks, improves classification performance across nearly all input settings. This demonstrates that even smaller models can achieve robust performance when guided by localized mask supervision, underscoring the efficiency and scalability of the OneN model family. Finally, when evaluating smaller baseline models such as MobileNet V3 and SqueezeNet, we observe limited classification performance. These architectures yield lower accuracy across all input settings, suggesting that their lightweight design, while efficient, may not be adequate for the complexity of the AnoGen dataset, especially in challenging settings such as *STD+BG*. ResNet-based architectures, although more capable than the lightweight models, still fall short of the performance achieved by our proposed OneN models.

Table 6 presents a detailed comparison of detection and localization performance across various models and XAI methods at the pixel level, on the AnoGen dataset. The OneN-B and OneN-L models demonstrate superior performance, consistently achieving the highest values across all metrics.

Table 6: Average pixel-level localization performance on the AnoGen dataset using the *STD+G* protocol. OneN results are reported with (☐) and without (✗) weak masks. Best and second-best results per column are bolded and underlined.

			Pixel-level		
Model	**XAI Method**	**Mask**	**AUC**	**AP**	**F_1-max**
ResNet-34	GradCAM	✗	56.72	3.81	7.67
ResNet-34	GradCAM++	✗	56.31	3.95	7.67
ResNet-34	ScoreCAM	✗	58.13	4.00	7.95
ResNet-34	LayerCAM	✗	56.30	3.89	7.63
ResNet-152	GradCAM	✗	50.67	4.57	8.42
ResNet-152	GradCAM++	✗	51.92	5.58	9.42
ResNet-152	ScoreCAM	✗	65.30	5.72	10.81
ResNet-152	LayerCAM	✗	51.76	4.66	8.59
ViT-B-16 (S)	Rollout	✗	79.49	12.63	18.74
ViT-B-16 (B)	Rollout	✗	84.18	14.62	21.64
ViT-B-16 (L)	Rollout	✗	82.74	12.54	19.16
CSKD-Ti	Rollout	✗	56.06	3.32	7.06
CSKD-S	Rollout	✗	61.10	3.53	7.34
CSKD-B	Rollout	✗	61.88	3.93	7.95
OneN-S	Rollout	✗	64.04	6.31	11.36
OneN-B	Rollout	✗	74.67	14.73	21.11
OneN-L	Rollout	✗	82.84	13.81	20.75
OneN-S	Rollout	☐	85.83	17.48	23.24
OneN-B	Rollout	☐	**88.61**	**20.90**	**27.47**
OneN-L	Rollout	☐	<u>88.55</u>	<u>19.05</u>	<u>26.02</u>

ViT-B-16 models also show comparable performance, suggesting that larger model capacity is not strictly necessary for effective localization. By contrast, the ResNet-based models (ResNet-34 and ResNet-152) produce similar results, with Score-CAM typically emerging as the best-performing XAI technique. This highlights a general limitation among these methods in generating fine-grained localization maps. The ViT-based models and the OneN family benefit significantly from the attention rollout technique, whereas CSKD architectures perform poorly across different metrics. We also observe that many methods suffer from low F_1-max and AP, primarily due to degraded precision and recall under class imbalance. Consistent with previous classification experiments, when weak annotations are provided, the performance of our OneN models, particularly when using attention rollout, improves substantially, especially in terms of AP and F_1-max. Finally, all models show similar image-level detection performance, with AUC, AP, and F_1-max averaging 43.69, 71.00, and 83.69, respectively, and varying by no more than 2%.

Table 7 presents the classification results on the AnoDiff dataset. Our

Table 7: Average classification performance on the AnoDiff dataset under TR-AW and T-AW strategies. OneN results are reported with no mask (✗), weakly-annotated mask (☐) and fine-grained mask (✓). Best and second-best scores per column are bolded and underlined, respectively.

		Accuracy (%)					
Model	**Mask**	***STD***	***STD+G***	***STD+G+BG***	***STD+BG***	***B***	***B+BG***
MobileNet V3	✗	19.80/35.05	17.41/38.15	22.58/40.17	16.24/44.11	46.95/57.92	27.46/52.77
SqueezeNet 1.0	✗	43.65/54.70	36.63/45.54	36.10/47.68	48.00/55.10	32.02/67.92	30.12/68.17
ResNet-34	✗	50.03/**69.30**	41.41/57.33	45.99/58.21	52.25/69.80	40.24/73.00	43.26/76.26
ResNet-152	✗	44.30/64.81	40.28/**60.08**	40.55/**64.18**	41.48/**73.60**	32.76/77.85	29.19/75.93
ViT-B-16 (S)	✗	40.76/49.84	34.03/42.66	29.87/38.31	41.77/50.30	28.51/66.14	28.56/72.17
ViT-B-16 (B)	✗	50.44/58.13	38.68/51.66	37.35/47.86	54.07/57.21	30.97/72.08	28.56/73.97
ViT-B-16 (L)	✗	50.45/58.15	38.45/50.28	34.63/46.14	55.90/59.31	28.07/65.52	28.66/69.60
CSKD-Ti	✗	20.23/23.92	22.73/25.71	20.21/23.61	27.83/30.56	49.39/71.82	45.56/67.39
CSKD-S	✗	20.52/24.33	20.37/24.26	20.75/25.83	28.86/31.71	47.53/67.79	**59.00**/68.75
CSKD-B	✗	24.52/24.76	23.77/33.20	17.11/28.74	29.10/32.12	40.61/72.18	**54.17**/72.74
OneN-S	✗	29.81/37.21	33.55/38.24	26.69/34.70	26.03/39.54	42.78/74.96	46.12/73.84
OneN-B	✗	43.37/51.17	40.79/48.27	32.08/43.75	34.94/50.43	44.64/76.34	42.43/75.32
OneN-L	✗	46.18/56.15	40.86/50.51	36.60/51.36	43.43/55.80	41.14/73.65	39.54/75.22
OneN-S	☐	31.70/37.81	31.34/36.20	19.47/34.99	35.43/39.64	43.58/73.25	31.56/72.80
OneN-B	☐	39.29/47.33	35.89/43.36	22.15/39.67	36.59/46.81	40.76/74.87	22.51/73.51
OneN-L	☐	43.83/56.31	41.38/52.18	39.14/47.01	44.09/53.48	39.54/71.53	30.81/73.77
OneN-S	✓	30.36/38.05	30.06/33.92	23.34/35.40	32.60/41.39	45.75/75.37	45.88/74.74
OneN-B	✓	43.10/53.92	38.80/49.95	33.36/48.64	40.14/58.72	44.62/73.49	42.82/**77.44**
OneN-L	✓	**50.97**/61.66	**47.12**/58.73	**50.86**/57.18	**56.88**/63.92	**49.55**/**78.84**	38.12/74.96

proposed OneN-L model achieves the best performance across nearly all input configurations under the fair evaluation protocol, namely TR-AW. This demonstrates that self-attention mechanisms can effectively capture subtle anomalous patterns from synthetic data. OneN-L particularly outperforms all competitors by a large margin, even when good and/or background samples are included in the classification task. In contrast, CSKD-based models consistently underperform across all settings, except in the *B+BG* configuration. We attribute this exception primarily to class imbalance issues in the dataset (see Table 4). Interestingly, ResNet-based architectures perform strongly under the T-AW evaluation strategy, even without leveraging additional masks, by achieving high classification accuracy or ranking as the second-best model across diverse input settings. This indicates that ResNet models possess strong generalization capabilities. However, their performance plateaus when trained using the TR-AW evaluation protocol, revealing a gap between the representational quality of synthetic samples and the complexity of the networks.

Table 8 reports both detection and localization performance at the pixel level across various models and XAI methods on the AnoDiff dataset. Our OneN model family achieves the best overall metrics. Specifically, OneN-L yields the highest AUC score when fine-grained annotations are used, while both OneN-B and OneN-L achieve the best AP and F_1-max values under weakly annotated masks. These results demonstrate that, even with weak supervision, our models are capable of effectively localizing industrial defects. In contrast, fine-grained annotations may negatively affect performance, potentially due to the lack of generalization learned during training. The ResNet-based models follow a similar trend observed in the AnoGen dataset, with Score-CAM providing the best performance, while CSKD models struggle due to limited supervision. At the image level, we observe similar performance across all methods, indicating comparable capabilities when localization is not required. As in the previous dataset, all models show similar image-level detection performance, with AUC, AP, and F_1-max averaging 48.79, 74.21, and 84.34, respectively, and varying by no more than 2%.

From a category-level perspective, we note that classification performance is not uniformly distributed across categories. Categories such as capsule, pill, and screw remain particularly challenging due to the presence of small or low-contrast defects, especially under multi-class settings such as *STD* and *STD+G*. For instance, the accuracy for these categories frequently re-

Table 8: Average pixel-level localization performance on the AnoDiff dataset using the *STD+G* protocol. OneN results are reported with no mask (✗), weakly-annotated mask (☐) and fine-grained mask (✓). Best and second-best results per column are bolded and underlined, respectively.

			Pixel-level		
Model	**XAI Method**	**Mask**	**AUC**	**AP**	F_1**-max**
ResNet-34	GradCAM	✗	57.74	5.42	9.98
ResNet-34	GradCAM++	✗	56.89	5.42	10.22
ResNet-34	LayerCAM	✗	57.18	5.48	10.33
ResNet-34	ScoreCAM	✗	59.62	6.07	10.92
ResNet-152	GradCAM	✗	57.00	6.45	11.49
ResNet-152	GradCAM++	✗	59.68	6.65	11.28
ResNet-152	LayerCAM	✗	59.77	6.47	11.45
ResNet-152	ScoreCAM	✗	64.30	7.06	12.41
ViT-B-16 (S)	|	✗	70.76	9.00	14.64
ViT-B-16 (B)	Rollout	✗	82.48	14.76	21.58
ViT-B-16 (L)	|	✗	78.61	12.11	18.93
CSKD-Ti	|	✗	54.62	4.08	7.61
CSKD-S	Rollout	✗	61.93	4.98	9.44
CSKD-B	|	✗	62.51	5.36	10.05
OneN-S	|	✗	59.22	6.98	12.21
OneN-B	|	✗	71.43	12.32	18.77
OneN-L	|	✗	75.84	13.11	20.07
OneN-S	|	☐	71.89	17.60	26.72
OneN-B	Rollout	☐	76.08	**20.90**	**28.97**
OneN-L	|	☐	75.38	<u>20.56</u>	<u>28.89</u>
OneN-S	|	✓	82.13	14.50	20.11
OneN-B	|	✓	<u>84.79</u>	14.45	21.74
OneN-L	|	✓	**85.04**	15.04	22.07

Table 9: Classification performance on the RAD dataset under TR-AW and T-AW strategies. OneN results are reported with (☐) and without (✗) weak masks. Best and second-best scores per column are bolded and underlined, respectively.

		Accuracy (%)					
Model	**Mask**	***STD***	***STD+G***	***STD+G+BG***	***STD+BG***	***B***	***B+BG***
MobileNet V3	✗	16.78/36.64	18.66/32.19	18.66/44.18	34.59/27.05	45.72/72.26	28.60/60.10
SqueezeNet 1.0	✗	19.86/20.83	12.50/50.34	36.30/11.64	23.97/20.89	48.80/41.95	59.76/69.01
ResNet-34	✗	12.33/23.29	20.21/27.40	11.82/15.24	15.75/15.41	45.03/58.05	15.07/10.79
ResNet-152	✗	23.63/24.32	17.47/25.68	25.51/27.40	30.14/29.45	58.56/37.33	34.59/25.34
ViT-B-16 (S)	✗	96.92/97.60	48.97/52.40	47.43/81.68	96.23/97.60	50.00/71.06	52.05/87.33
ViT-B-16 (B)	✗	99.32/**100.0**	51.88/61.64	49.14/70.72	99.32/99.32	52.74/72.26	50.00/81.51
ViT-B-16 (L)	✗	99.32/**100.0**	48.80/92.64	48.12/74.66	97.95/**100.0**	50.00/70.89	49.83/71.23
CSKD-Ti	✗	48.97/56.51	18.66/48.80	54.62/55.65	42.81/49.66	52.05/73.80	55.48/56.16
CSKD-S	✗	37.67/57.53	35.45/53.77	23.12/89.04	52.74/88.70	50.00/79.11	50.68/52.57
CSKD-B	✗	96.92/96.58	**78.94**/90.58	56.51/65.07	97.26/94.18	50.00/76.71	55.48/56.16
OneN-S	✗	93.49/95.21	67.29/68.49	55.14/58.56	94.52/95.55	50.00/56.16	50.00/62.33
OneN-B	✗	98.63/99.66	50.34/77.74	47.95/**97.95**	99.32/**100.0**	52.05/82.19	50.68/56.85
OneN-L	✗	98.97/**100.0**	49.66/87.50	48.46/97.26	99.32/**100.0**	50.00/81.16	50.00/69.35
OneN-S	☐	94.18/95.21	59.08/72.26	73.63/87.16	97.95/98.29	50.00/60.27	50.00/63.01
OneN-B	☐	98.97/**100.0**	74.14/**99.14**	**77.91**/93.84	96.92/99.66	73.29/93.15	**80.82**/99.32
OneN-L	☐	**100.0**/**100.0**	63.87/96.23	67.98/80.14	**100.0**/**100.0**	51.37/80.14	51.20/**100.0**
OneN-S	✓	95.89/99.32	73.63/89.90	62.33/91.10	97.60/97.95	50.68/72.60	50.00/56.85
OneN-B	✓	99.66/**100.0**	73.80/83.73	66.78/94.35	99.66/99.66	**78.08**/91.78	77.40/95.21
OneN-L	✓	**100.0**/**100.0**	51.37/88.01	49.83/94.52	99.66/99.66	50.00/**97.09**	50.00/82.02

mains below 30%. This limitation is partially alleviated in the *B* and *B+BG* configurations, where combining all defect types into a single class simplifies the classification task. By contrast, categories characterized by visually distinct or structurally regular features, such as hazelnut, tile, metal nut, and transistor, consistently show high and stable accuracy across most input settings. Furthermore, a direct comparison between the two datasets highlights notable differences in performance trends. On average, AnoGen achieves superior results in configurations that include background and binary groupings (*B*, *B+BG*), with several challenging categories (*e.g.*, grid, carpet, and tile) showing marked improvements in accuracy. By contrast, AnoDiff demonstrates more stable performance under the standard multi-class settings (*STD*, *STD+G*), especially in object-centric categories such as bottle, transistor, and metal nut.

Table 9 presents the classification accuracies on the RAD dataset. This dataset introduces real-world imaging challenges, including viewpoint variation, lighting inconsistencies, and motion blur, which together make classification particularly difficult. Despite these complexities, our OneN framework

Table 10: Average pixel-level localization performance on the RAD dataset using the *STD+G* protocol. OneN results are reported with (☐) and without (✗) weak masks. Best and second-best results per column are bolded and underlined.

			Pixel-level		
Model	**XAI Method**	**Mask**	**AUC**	**AP**	**F_1-max**
ResNet-34	GradCAM	✗	69.60	5.94	11.48
ResNet-34	GradCAM++	✗	82.76	12.28	20.77
ResNet-34	ScoreCAM	✗	82.38	12.25	20.36
ResNet-34	LayerCAM	✗	83.27	13.08	21.69
ResNet-152	GradCAM	✗	51.64	3.00	5.94
ResNet-152	GradCAM++	✗	76.51	6.88	12.98
ResNet-152	ScoreCAM	✗	78.17	7.00	13.20
ResNet-152	LayerCAM	✗	76.59	6.64	12.55
ViT-B-16 (S)	|	✗	65.98	6.28	12.84
ViT-B-16 (B)	Rollout	✗	77.12	9.24	16.04
ViT-B-16 (L)	|	✗	74.52	12.58	21.90
CSKD-Ti	|	✗	73.09	11.29	22.37
CSKD-S	Rollout	✗	69.79	11.09	20.44
CSKD-B	|	✗	87.15	16.42	25.60
OneN-S	|	✗	85.29	15.91	23.41
OneN-B	|	✗	81.35	9.99	19.05
OneN-L	|	✗	83.21	11.37	20.91
OneN-S	|	☐	72.30	24.02	<u>32.99</u>
OneN-B	Rollout	☐	69.33	22.45	30.86
OneN-L	|	☐	71.31	<u>22.87</u>	32.18
OneN-S	|	✓	**92.94**	**43.97**	**47.78**
OneN-B	|	✓	<u>87.24</u>	14.91	23.82
OneN-L	|	✓	85.68	13.38	21.97

consistently achieves the highest overall performance, ranking first or second across nearly all input configurations. When equipped with weak mask supervision, the OneN models achieve the highest accuracy under several settings, highlighting the effectiveness of localized guidance without requiring a large number of parameters. We also observe that smaller variants of our model, such as OneN-S, demonstrate strong robustness, highlighting the scalability and efficiency of the OneN family. In contrast to their performance on the AnoGen dataset, ViT models perform considerably better on RAD, frequently outperforming CNN-based baselines. The CSKD family exhibits more mixed results, with performance varying across configurations. While CNN baselines show some improvement under the T-AW evaluation protocol, their overall performance remains limited, further emphasizing the challenges posed by the RAD dataset and the constraints of lightweight architectures in such complex scenarios.

Table 10 reports the average pixel-level localization performance on the

Table 11: Classification performance on the MTSD dataset under TR-AW and T-AW strategies. OneN results are reported with no mask (✗), weakly-annotated mask (☐) and fine-grained mask (✓). Best and second-best scores per column are bolded and underlined, respectively.

		Accuracy (%)					
Model	**Mask**	***STD***	***STD+G***	***STD+G+BG***	***STD+BG***	***B***	***B+BG***
MobileNet V3	✗	25.00/21.88	34.43/8.68	12.57/22.16	19.79/30.21	71.26/68.26	29.04/12.87
SqueezeNet 1.0	✗	25.00/11.46	17.07/29.04	17.66/45.51	25.00/27.08	71.26/52.10	27.54/69.46
ResNet-34	✗	18.75/32.29	57.49/21.26	12.28/45.51	26.04/30.21	32.63/65.57	14.07/21.56
ResNet-152	✗	25.00/36.46	28.74/22.46	13.77/18.56	20.83/34.38	44.91/64.07	46.11/68.26
ViT-B-16 (S)	✗	82.29/89.58	71.26/71.56	71.26/77.84	90.62/90.62	71.26/72.46	71.26/71.56
ViT-B-16 (B)	✗	95.83/<u>97.92</u>	78.44/<u>89.52</u>	79.34/**86.83**	<u>97.92</u>/<u>97.92</u>	**88.32**/**89.52**	79.04/83.83
ViT-B-16 (L)	✗	<u>94.79</u>/**98.96**	**89.22**/89.22	<u>81.74</u>/<u>84.73</u>	94.79/**98.96**	**88.32**/87.72	<u>82.04</u>/84.73
CSKD-Ti	✗	26.04/27.08	71.26/71.26	71.26/71.26	27.08/29.17	71.26/71.26	71.26/71.26
CSKD-S	✗	28.12/28.12	71.26/71.26	71.26/71.26	19.79/25.00	28.74/71.26	71.26/71.26
CSKD-B	✗	30.21/29.17	71.26/71.26	71.26/71.26	26.04/26.04	71.26/71.26	71.26/71.26
OneN-S	✗	53.12/53.12	71.26/71.26	70.36/71.26	50.00/54.17	71.26/71.26	70.96/71.26
OneN-B	✗	91.67/96.88	71.26/76.65	77.54/80.24	93.75/98.96	71.26/81.14	78.14/83.83
OneN-L	✗	**97.92**/**98.96**	82.34/87.72	**86.23**/**86.83**	95.83/**98.96**	83.23/87.72	**83.53**/**88.02**
OneN-S	☐	30.21/33.33	71.26/71.26	70.66/71.26	30.21/34.38	71.26/71.26	70.96/72.46
OneN-B	☐	39.58/42.71	71.26/71.26	71.26/75.15	79.17/76.04	71.26/71.56	70.96/71.86
OneN-L	☐	86.46/**98.96**	<u>82.93</u>/87.43	80.84/83.23	**98.96**/**98.96**	85.63/85.93	81.74/85.33
OneN-S	✓	25.00/36.46	71.26/71.26	70.36/71.26	34.38/35.42	71.26/71.26	70.66/71.56
OneN-B	✓	77.08/94.79	70.96/73.05	74.85/77.84	90.62/95.83	71.26/79.64	76.65/84.13
OneN-L	✓	87.50/**98.96**	82.04/**89.82**	78.74/83.53	96.88/**98.96**	<u>85.93</u>/<u>88.02</u>	81.44/<u>86.23</u>

RAD dataset. Our OneN framework demonstrates strong localization performance, especially when guided by class-specific mask supervision. OneN-S achieves the best performance across all three metrics when trained with full mask guidance, outperforming all other models. Among the mask-augmented configurations, OneN variants consistently surpass their corresponding non-masked counterparts, demonstrating the critical role of even weak spatial supervision in improving localization quality. OneN-S with partial mask supervision attains the second-best F_1-max, while OneN-L achieves the second-best AP, further demonstrating the effectiveness of targeted guidance for anomaly localization. Among the baseline models, CSKD-B shows the strongest localization performance in the absence of mask supervision. Finally, all models show similar image-level detection performance, with AUC, AP, and F_1-max averaging 34.92, 41.21, and 66.67, respectively, and varying by no more than 1%.

Table 11 reports the classification accuracies on the MTSD dataset. Despite the inherent challenges of this dataset, the proposed OneN-L model consistently achieves superior performance, ranking first or second across the majority of configurations and evaluation protocols. Notably, even in the absence of mask supervision, OneN-L outperforms all competing models, highlighting its robustness and strong generalization capabilities. The OneN-B variant also demonstrates competitive performance, particularly under the *B+BG* and *STD+BG* settings. The lightweight OneN-S model offers an effective balance between accuracy and computational efficiency, maintaining stable performance across various configurations. Transformer-based ViT models also exhibit strong results, particularly when compared to the smaller OneN variants, occasionally surpassing them in specific settings. In contrast, the CSKD models tend to exhibit consistent yet less adaptable performance across different strategies. Conventional CNN baselines, such as MobileNet V3 and SqueezeNet, generally underperform relative to both transformer-based and OneN models, underscoring the advantages of the OneN architecture in addressing the complexities of the MTSD dataset. Finally, all models show similar image-level detection performance, with AUC, AP, and F_1-max averaging 48.97, 26.90, and 45.11, respectively, and varying by no more than 2%.

Table 12 reports the pixel-level localization performance on the MTSD dataset using the *STD+G* protocol. The OneN models achieve superior localization performance compared to all baselines. For example, OneN-L, when trained with weak mask supervision, obtains the best scores for the AP and F_1-max metrics. OneN-B with weak supervision also performs strongly, ranking second in AP and F_1-max. Interestingly, the performance of OneN-B and OneN-L drops only slightly when trained without masks, indicating that the OneN framework is inherently capable of capturing localized anomalies even in the absence of explicit spatial guidance. Among transformer baselines, ViT-B-16 (L) achieves the best results. In contrast, CAM-based visual explanation methods applied to ResNet variants consistently yield poor localization performance, with low AUC values. Nevertheless, absolute AP and F_1-max values remain relatively low. This highlights the intrinsic difficulty of the MTSD dataset, which involves localizing fine-grained, low-contrast surface defects under variable illumination and complex backgrounds. In such scenarios, saliency-based methods often struggle to produce precise and dense activations, especially in the absence of a supervision.

Table 13 presents the average classification performance on the PCB dataset. Overall, OneN-L achieves the best results, obtaining the highest average accuracies across most evaluation protocols. This demonstrates its strong capacity to leverage weak supervision for improved classification. Even

Table 12: Average pixel-level localization performance on the MTSD dataset using the *STD+G* protocol. OneN results are reported with no mask (✗), weakly-annotated mask (☐) and fine-grained mask (✓). Best and second-best results per column are bolded and underlined.

			Pixel-level		
Model	**XAI Method**	**Mask**	**AUC**	**AP**	**F_1-max**
ResNet-34	GradCAM	✗	52.87	2.60	5.00
ResNet-34	GradCAM++	✗	56.73	2.99	5.34
ResNet-34	ScoreCAM	✗	57.74	3.12	6.05
ResNet-34	LayerCAM	✗	56.92	2.98	5.41
ResNet-152	GradCAM	✗	47.44	2.35	4.37
ResNet-152	GradCAM++	✗	49.53	2.47	4.58
ResNet-152	ScoreCAM	✗	49.99	2.52	4.83
ResNet-152	LayerCAM	✗	49.98	2.53	4.88
ViT-B-16 (S)	|	✗	53.41	2.45	4.69
ViT-B-16 (B)	Rollout	✗	55.16	2.87	5.43
ViT-B-16 (L)	|	✗	56.95	2.91	5.82
CSKD-Ti	|	✗	52.37	2.73	5.33
CSKD-S	Rollout	✗	53.28	2.80	5.52
CSKD-B	|	✗	50.57	2.55	5.04
OneN-S	|	✗	53.04	2.41	4.76
OneN-B	|	✗	**65.26**	3.85	7.62
OneN-L	|	✗	57.85	2.84	5.51
OneN-S	|	☐	47.23	5.16	9.16
OneN-B	Rollout	☐	51.06	<u>5.56</u>	<u>10.16</u>
OneN-L	|	☐	56.58	**6.39**	**11.88**
OneN-S	|	✓	48.12	2.32	4.48
OneN-B	|	✓	59.91	2.94	5.71
OneN-L	|	✓	<u>62.54</u>	3.13	6.49

without mask supervision, OneN-L maintains robust performance, frequently ranking as the second-best model. OneN-B also performs competitively, particularly in the binary classification setting. Despite its compact architecture, OneN-S delivers solid performance, often outperforming the CSKD variants, thus providing an efficient yet effective solution.

Among transformer-based baselines, ViT models exhibit strong performance, with larger variants such as ViT-B (L) and ViT-B (S) occasionally surpassing smaller OneN models in specific configurations. This highlights the representational power of transformer architectures in modeling complex defect patterns. In contrast, CSKD models yield lower but stable performance across training settings, suggesting limited adaptability to the unique challenges of the PCB dataset. Conventional CNN baselines underperform relative to both the OneN models and the transformer-based baselines.

Table 14 reports pixel-level localization performance on the PCB dataset

Table 13: Average classification performance on the PCB dataset under TR-AW and T-AW strategies. OneN results are reported with no mask (✗) and weakly-annotated mask (☐). Best and second-best scores per column are bolded and underlined, respectively.

		Accuracy (%)					
Model	**Mask**	***STD***	***STD+G***	***STD+G+BG***	***STD+BG***	***B***	***B+BG***
MobileNet V3	✗	19.94/21.05	16.15/16.63	8.79/17.34	15.24/15.51	52.08/52.91	33.24/44.46
SqueezeNet 1.0	✗	17.17/13.02	11.16/14.73	6.18/14.73	20.50/12.19	53.32/49.31	43.91/41.41
ResNet-34	✗	20.50/15.51	16.39/16.63	12.11/14.49	14.96/14.96	50.00/51.66	24.52/43.63
ResNet-152	✗	16.34/20.50	13.78/13.78	8.08/13.30	12.47/13.85	51.25/50.42	24.38/35.87
ViT-B-16 (S)	✗	19.67/23.82	18.29/17.81	22.80/23.28	16.34/25.48	63.43/65.10	61.36/<u>68.01</u>
ViT-B-16 (B)	✗	18.28/27.15	16.63/20.19	18.76/21.62	21.05/26.59	64.68/65.24	50.00/66.62
ViT-B-16 (L)	✗	19.11/26.04	19.71/20.90	22.80/27.55	17.17/24.65	62.47/68.01	<u>68.14</u>/64.27
CSKD-Ti	✗	22.71/22.16	10.69/17.34	14.25/14.49	22.16/21.33	49.72/57.62	62.60/61.50
CSKD-S	✗	22.16/12.74	14.73/19.00	14.01/14.25	22.16/22.44	50.28/50.14	49.58/55.40
CSKD-B	✗	12.47/22.16	14.25/18.76	13.78/16.39	22.16/13.85	55.26/60.39	60.39/62.19
OneN-S	✗	22.16/24.93	13.30/14.25	13.78/19.71	14.96/21.88	62.60/<u>67.17</u>	61.91/65.24
OneN-B	✗	19.94/22.44	15.44/16.86	16.86/21.38	22.44/24.93	61.36/65.51	59.28/66.07
OneN-L	✗	<u>29.92</u>/<u>36.57</u>	<u>27.55</u>/<u>28.74</u>	**35.63**/<u>30.64</u>	<u>37.12</u>/<u>36.29</u>	53.46/66.07	64.82/67.73
OneN-S	☐	16.90/23.27	17.81/14.73	14.96/22.09	21.05/24.93	63.43/64.82	61.63/64.13
OneN-B	☐	16.34/24.93	17.81/16.86	16.15/22.57	18.56/26.59	<u>66.76</u>/64.68	63.30/67.04
OneN-L	☐	**49.31/50.97**	**29.69/32.07**	<u>34.68</u>/**40.14**	**46.26/52.08**	**66.90/68.56**	**69.39/68.56**

using the *STD+G* protocol. The proposed OneN models outperform all baselines in this setting. Specifically, when trained with weak mask supervision, OneN-L achieves the highest scores across all metrics, including AUC, AP, and F_1-max. Similarly, OneN-S and OneN-B under weak supervision also rank among the top-performing methods, with OneN-S obtaining the second-highest F_1-max. Even without mask supervision, OneN models still localize effectively fine-grained defects without explicit pixel-level guidance. Among transformer-based models, ViT-B-16 (L) performs best, although it still lags behind OneN-L. In contrast, conventional CNN models employing CAM-based methods on ResNet backbones exhibit comparatively lower localization accuracy. These methods show limited sensitivity to subtle defect cues, likely due to the coarse activation maps. Nonetheless, despite these relative improvements, the absolute AP and F_1-max values remain low across all models. This highlights the intrinsic difficulty of the PCB dataset, where defects tend to be small and localized, making precise spatial localization challenging even for advanced architectures. Finally, all models show similar image-level detection performance, with AUC, AP, and F_1-max averaging 44.91, 84.05, and 92.33, respectively, and varying by no more than 1%.

Qualitative Results. Figure 4 presents a qualitative comparison of localization performance across the proposed OneN networks trained on the

Table 14: Average pixel-level localization performance on the PCB dataset using the *STD+G* protocol. OneN results are reported with no mask (✗) and weakly-annotated mask (☐). Best and second-best results per column are bolded and underlined, respectively.

			Pixel-level		
Model	**XAI Method**	**Mask**	**AUC**	**AP**	$\mathbf{F_1}$**-max**
ResNet-34	GradCAM	✗	56.94	2.26	4.44
ResNet-34	GradCAM++	✗	56.43	2.28	4.51
ResNet-34	ScoreCAM	✗	55.17	2.11	4.35
ResNet-34	LayerCAM	✗	56.39	2.23	4.44
ResNet-152	GradCAM	✗	52.38	1.95	3.96
ResNet-152	GradCAM++	✗	56.39	2.20	4.31
ResNet-152	ScoreCAM	✗	56.87	2.25	4.42
ResNet-152	LayerCAM	✗	56.57	2.21	4.33
ViT-B-16 (S)	|	✗	52.36	2.00	3.90
ViT-B-16 (B)	Rollout	✗	46.36	1.62	3.70
ViT-B-16 (L)	|	✗	55.34	2.19	4.34
CSKD-Ti	|	✗	51.92	1.99	3.78
CSKD-S	Rollout	✗	51.03	1.90	3.81
CSKD-B	|	✗	53.61	2.01	3.95
OneN-S	|	✗	44.75	1.55	3.69
OneN-B	|	✗	48.27	1.82	3.67
OneN-L	|	✗	46.53	1.63	3.67
OneN-S	|	☐	<u>58.47</u>	2.61	<u>5.36</u>
OneN-B	Rollout	☐	54.82	2.40	4.98
OneN-L	|	☐	**63.82**	**2.85**	**5.73**

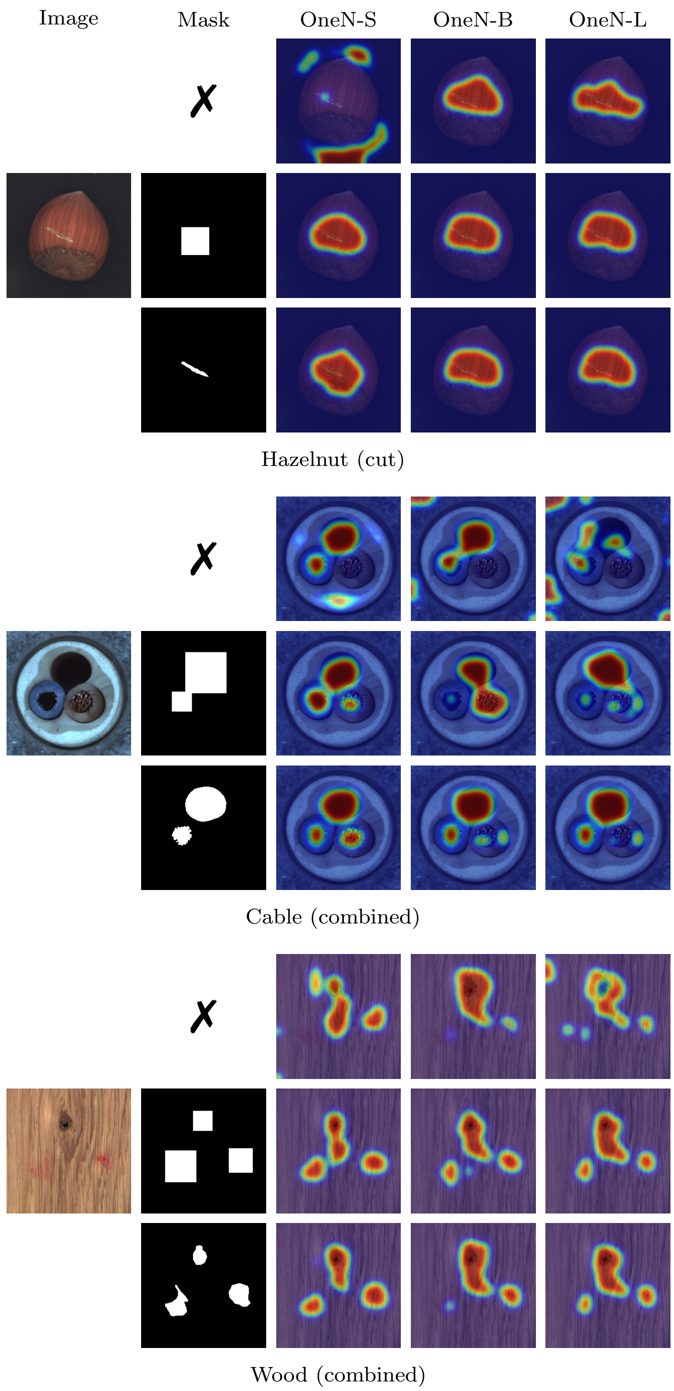


Figure 4: Qualitative localization results of OneN models on the AnoDiff dataset with three supervision settings: no masks (1st row), weak masks (2nd row), and ground-truth masks (3rd row). Best viewed in color.

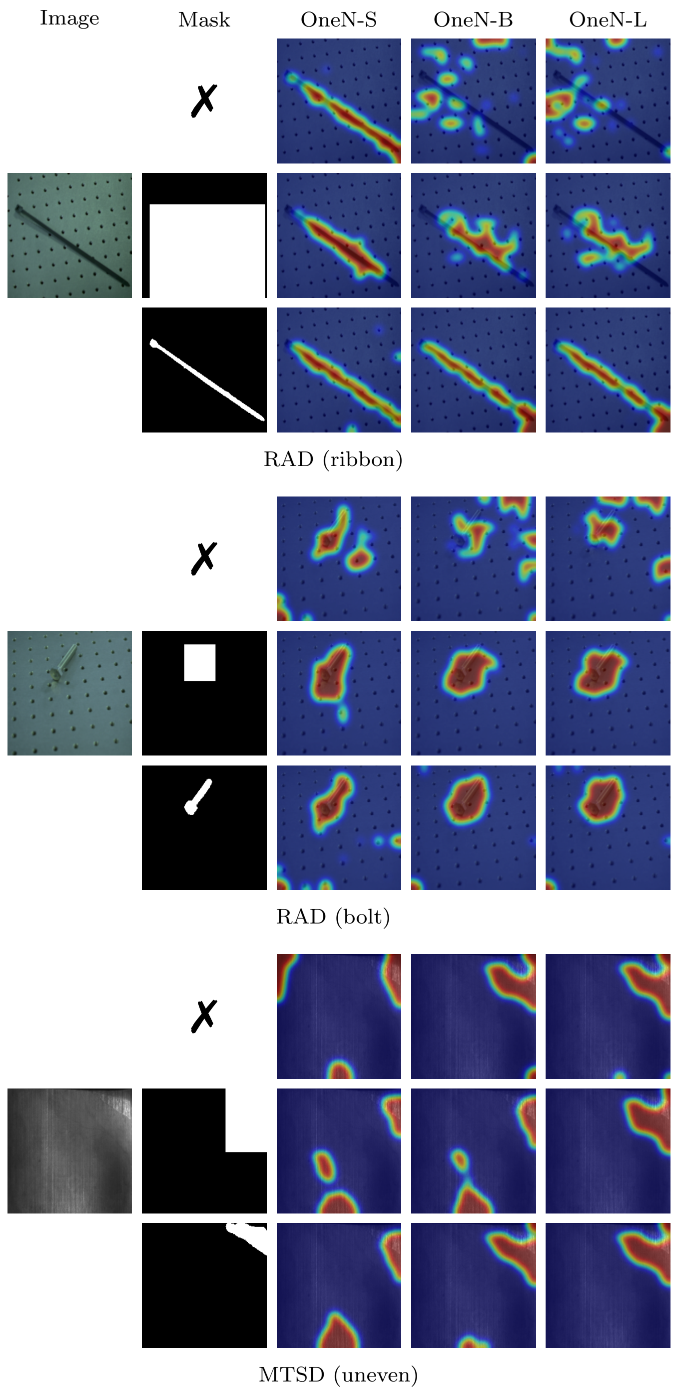


Figure 5: Qualitative localization results of OneN models on the RAD and MTSD datasets with three supervision settings: no masks (1st row), weak masks (2nd row), and ground-truth masks (3rd row). Best viewed in color.

AnoDiff dataset under different supervision levels. OneN-S is able to identify the anomaly regions but tends to slightly under-localize and produces less focused activation maps, particularly when no additional masks or only weakly annotated masks are used. OneN-B yields more accurate localization, better aligning with the shape and position of the ground-truth masks. In contrast, OneN-L delivers the most precise and confident localization, suggesting that it benefits the most from its increased capacity, achieving strong results even in the absence of fine-grained supervision. Our models are also capable of localizing multiple defects, such as compound anomalies in the wood or cable categories, by activating multiple regions corresponding to the defective areas. These results highlight that weak annotations are beneficial in most cases, enabling effective localization while significantly reducing the cost and effort associated with labor-intensive annotation processes.

Figure 5 also illustrates localization performance of our OneN variants on samples from the RAD and MTSD datasets. In the RAD dataset, for both ribbon and bolt defect types, OneN-S generally succeeds in detecting anomalous regions but often show fragmented or overly diffuse activation maps. OneN-B shows better performance, especially for subtle defects, by producing more structured and contiguous localization aligned with defect morphology. While OneN-S presents better localization capability for the ribbon class, OneN-L and OneN-B provide more accurate localization for the bolt class. Similar trends are observed in the MTSD dataset, where OneN-B delivers the most confident and spatially coherent localization, accurately delineating uneven texture regions.

Figure 6 presents qualitative localization results of the proposed OneN models on the PCB dataset, focusing on two representative defect types. In the missing hole example, all models successfully activate around the defect location, but the spatial extent and sharpness of the activations vary. OneN-S produces relatively broad activation maps with weaker central focus, while OneN-B shows more concentrated responses, though it occasionally over-activates nearby non-defective regions. OneN-L produces the most precise and sharply defined heatmaps, closely aligned with the actual defect location, even in the absence of masks. When trained with weak supervision, both OneN-B and OneN-L demonstrate improved localization. In the more complex short defect case, localization becomes more challenging. Without supervision, the OneN models tend to activate across multiple surrounding regions, often missing the precise defect boundaries. In contrast, guided training improves localization accuracy, resulting in activations that are more

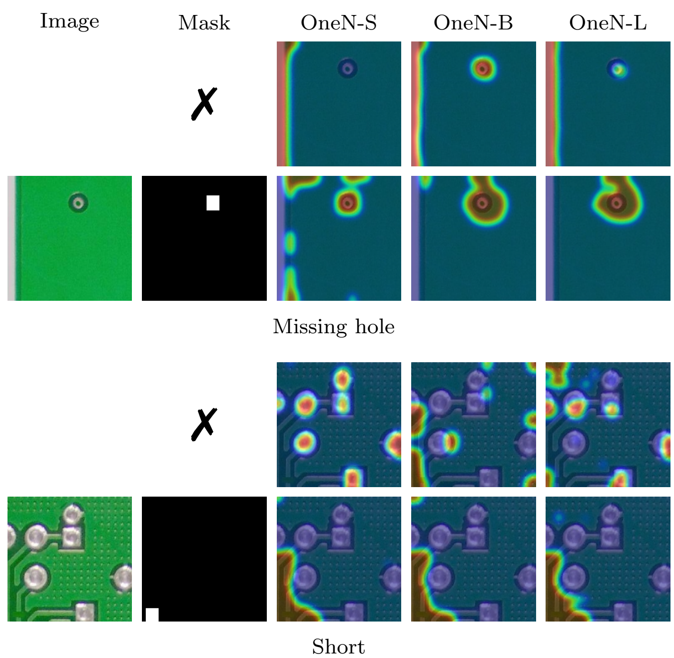


Figure 6: Qualitative localization results of OneN models on the PCB dataset with two supervision settings: no masks (1st row) and weak masks (2nd row). Best viewed in color.

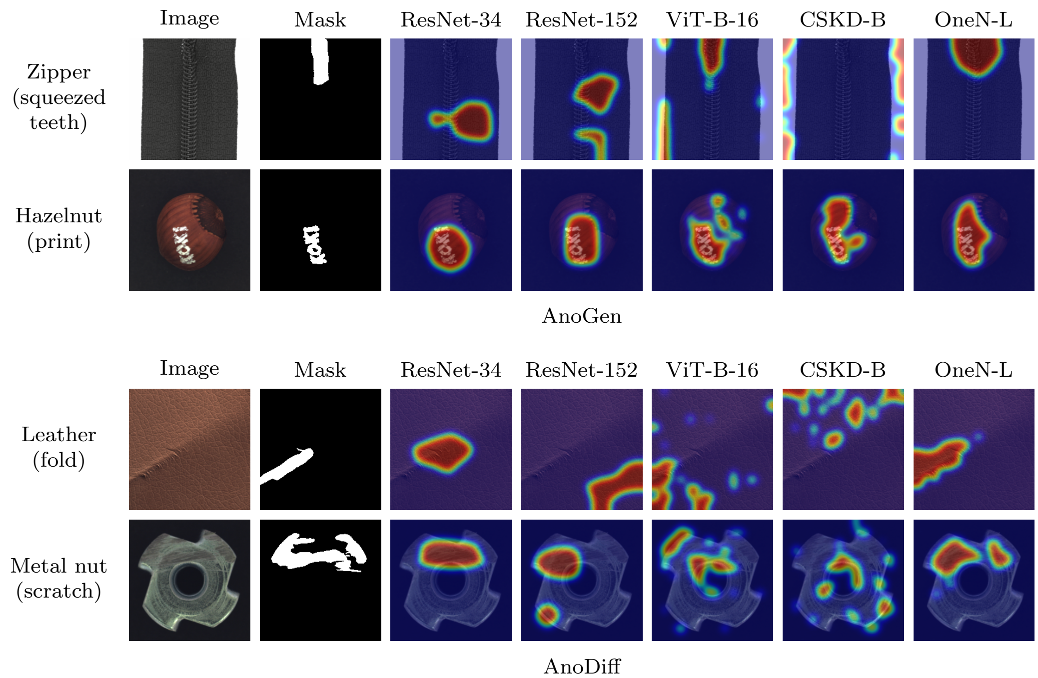


Figure 7: Qualitative localization comparison on AnoGen (1^{st}, 2^{nd} rows) and AnoDiff (3^{rd}, 4^{th}) using our OneN-L model , alongside CNN and ViT-based models. GradCAM++ and attention rollout are used for generating heatmaps from CNNs and attention-based models, respectively. Best viewed in color.

tightly confined to the true defect regions, suggesting enhanced feature discrimination capabilities.

Figure 7 provides a visual comparison of localization performance among ResNet-based models, ViT-B-16 (L), CSKD-B, and our OneN-L network. The results show that OneN-B consistently produces sharper and more focused localization maps, aligning more accurately with the ground-truth masks across both datasets. While ResNet-34 and ResNet-152 are able to highlight some anomalous regions, their responses tend to be less precise and occasionally fragmented. For instance, ResNet-34 struggles to identify complex anomaly patterns, often activating fixed, sharply defined regions that do not correspond to actual defects. In contrast, OneN-L benefits from its tailored architecture and mask-based supervision, resulting in more reliable and interpretable localization. CSKD-B appears unable to effectively detect defective regions in complex objects. For example, this network mistakenly identifies the black border of a zipper object as anomalous, and produces

Table 15: Comparison of various PMFL loss configurations evaluated on the wood category of the AnoDiff dataset using OneN-L trained with fine-grained annotations. Best and second-best results per column are bolded and underlined, respectively.

Loss Params.			Pixel-level	
β	Progressive	Acc. (%)	AUC	AP
1	✗	75.95	31.67	66.09
1	✓	72.15	25.35	62.87
3	✗	70.89	**41.05**	68.64
3	✓	74.68	32.63	66.35
7	✗	75.95	35.13	67.72
7	✓	**78.48**	40.18	**71.38**
10	✗	77.22	31.23	65.55
10	✓	74.68	22.81	62.70

scattered or inconsistent results for other categories.

Ablation studies. In the following, we report both quantitative and qualitative ablation studies to evaluate the robustness of our family models. More specifically, Table 15 provides a comparison of different configurations of the proposed PMFL loss, examining the influence of the weighting parameter β and the presence of the progressive mechanism. Our experiments reveal that the configuration with $\beta = 7$ and the progressive mechanism yields the best overall performance, mainly in terms of accuracy and AP pixel-level, while also ranking second-best in pixel-level AUC.

Table 16 presents the classification results for both datasets, evaluated across several categories independently. Our OneN-L model typically achieves the best performance under both evaluation protocols. Interestingly, we observe that additional input annotations are particularly useful for object-based categories, *e.g.*, hazelnut, whereas texture-based categories often perform comparably well even without additional supervision. This suggests that weakly or finely annotated inputs provide more discriminative power in structured, patterned images. In several cases, the OneN-B model benefits from fine-grained annotations, achieving either the best or second-best performance. The OneN-S model exhibits stable performance under the TR-AW strategy regardless of annotation type, but shows greater sensitivity to annotations under the T-AW protocol. These results confirm the robustness of our model family across different supervision levels and input types.

Figure 8 shows a qualitative comparison corresponding to the experiments reported in Table 16 using both datasets. We observe that the OneN-B and OneN-L models produce correct classification predictions, with attention

Table 16: Classification results (%) on AnoGen ((a), (b)) and AnoDiff ((c), (d)) datasets for various categories under the *STD + G* input, evaluated with both TR-AW and T-AW strategy. Best and second-best per column are bolded and underlined, respectively.

	Mask	
Model	✗	□
OneN-S	45.30/50.43	48.72/51.28
OneN-B	86.32/**91.45**	77.78/88.89
OneN-L	**88.03**/90.60	**87.18**/**94.02**

(a) Tile

	Mask	
Model	✗	□
OneN-S	46.84/58.23	60.76/68.35
OneN-B	67.09/**78.48**	**79.75**/**78.48**
OneN-L	**69.62**/**78.48**	74.68/75.95

(b) Hazelnut

	Mask		
Model	✗	□	✓
OneN-S	53.64/50.91	52.73/52.73	49.09/62.73
OneN-B	44.55/**74.55**	73.64/72.73	51.82/**84.55**
OneN-L	**60.00**/67.27	**77.27**/**81.82**	**63.64**/80.91

(c) Wood

	Mask		
Model	✗	□	✓
OneN-S	27.35/26.50	22.22/25.64	25.64/30.77
OneN-B	42.74/44.44	**35.90**/50.43	47.01/**52.14**
OneN-L	**58.97**/**48.72**	31.62/**58.97**	**50.43**/**52.14**

(d) Carpet

maps accurately focusing on the defective image regions. Although OneN-S makes an incorrect prediction for a sample image (see Figure 8a), it still successfully highlights most of the defective regions. In Figure 8b, OneN-S provides a correct prediction but yields more scattered activation maps, failing to clearly identify the defective areas. In both Figure 8c and Figure 8d, all OneN variants correctly identify the defects, even when they are distributed across multiple regions of the image.

Figure 9 shows some representative failure cases of the OneN-L model's defect localization performance across diverse industrial datasets. Each example highlights specific challenges encountered during anomaly detection, such as difficulty in precisely delineating subtle, linear defects within patterned backgrounds (see Figure 9a), or over-segmentation leading to broader detection on simpler objects (see Figure 9b). The model also shows difficulty with irregular, distributed defects (see Figure 9c), where localization is diffuse, and with accurately capturing slender, elongated anomalies (see Figure 9d). Furthermore, we notice a recurring issue of over-extending the localization area beyond the actual defect boundary (see Figure 9e) and struggling to isolate small, distinct defects within complex, densely packed structures (see Figure 9f).

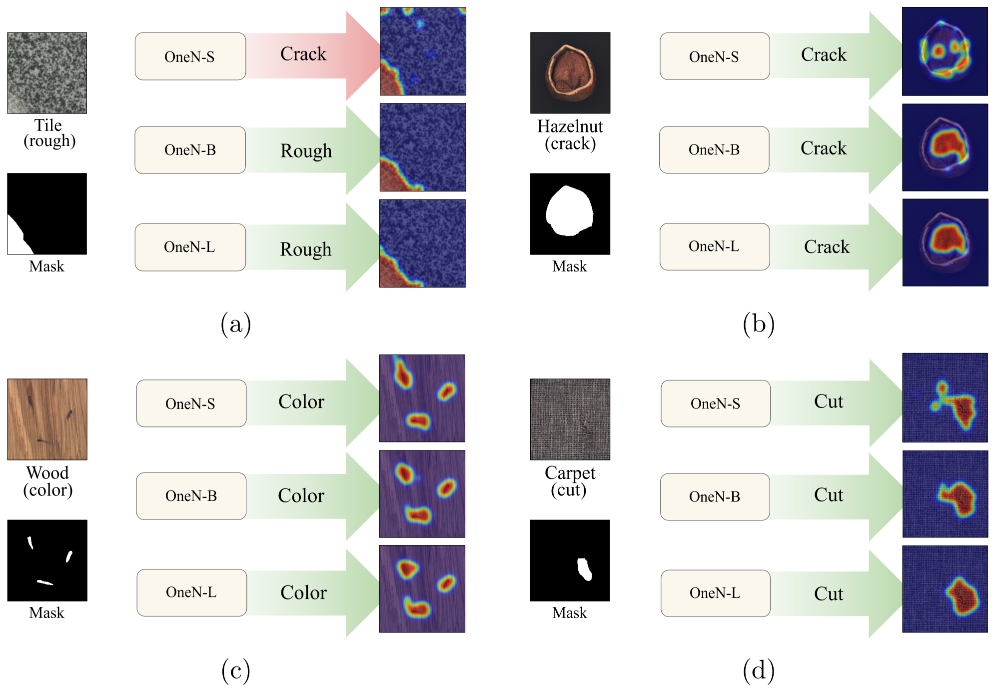
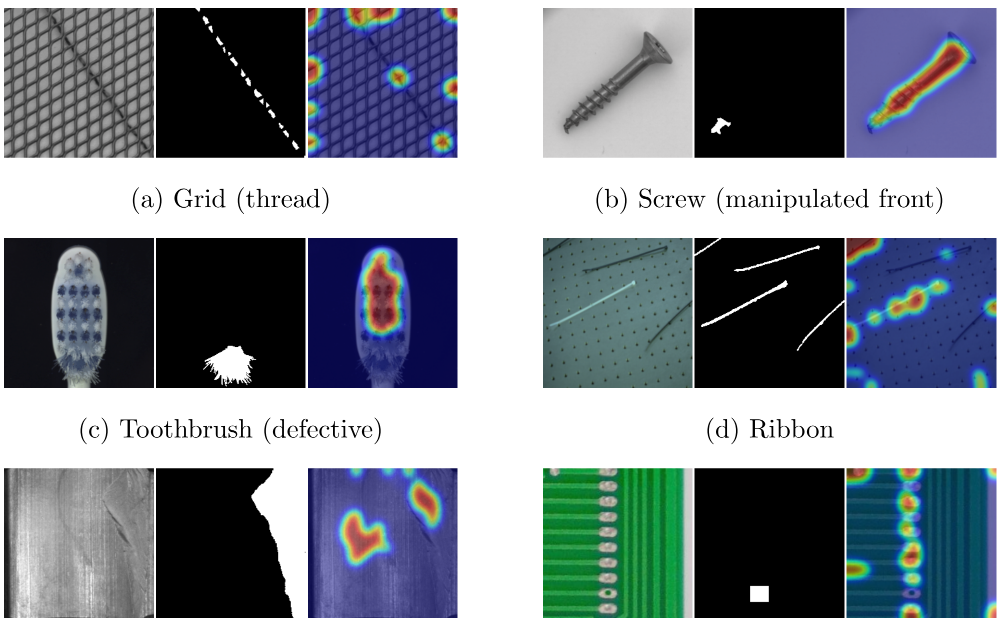


Figure 8: Classification and localization results of OneN models on the weakly supervised AnoGen dataset ((a), (b)) and the fully supervised AnoDiff dataset ((c), (d)). Classification outputs are color-coded (red: incorrect, green: correct). Best viewed in color.

5. Conclusion

Anomaly detection represents a fundamental task across various industrial domains, yet it remains challenged by factors such as human error, processing bottlenecks, limited adaptability of conventional deep learning approaches, and the scarcity of defective samples necessary for training robust models. In this work, we propose a unified architecture capable of performing both classification and detection/localization, tasks that are conventionally addressed by separate models. Our method employs knowledge distillation to transfer representational knowledge from a high-capacity convolutional neural network to attention-based models trained under different supervision regimes. To enhance localization accuracy, we introduce a progressive masking mechanism that effectively guides attention maps toward precise defect regions. Comprehensive experimental evaluations demonstrate that the proposed approach achieves strong performance across diverse industrial scenarios and input configurations, even under weak supervision.



(a) Grid (thread)

(b) Screw (manipulated front)

(c) Toothbrush (defective)

(d) Ribbon

(e) Break

(f) Missing hole

Figure 9: Failure cases in localization maps from our OneN-L model trained on the considered datasets: (a) and (b) AnoDiff, (c) AnoGen, (d) RAD, (e) MTSD, and (f) PCB. Best viewed in color.

This reduces computational overhead, and substantially decreases reliance on manual annotation.

Future research will focus on enhancing the efficiency and deployability of the OneN framework in industrial scenarios. Specifically, we plan to further investigate class imbalance across various anomaly detection tasks and explore quantization techniques to enable deployment on low-power and embedded hardware. Another promising direction involves distilling knowledge into multiple compact variants by reducing network depth and patch resolution, thereby offering lightweight yet interpretable solutions suitable for diverse edge devices.

Acknowledgments

This work was supported in part by the EC under projects Chips JU EdgeAI (101097300) and by project SERICS (PE00000014) under the MUR NRRP funded by the EU-NGEU. Project EdgeAI is supported by the Chips Joint Undertaking and its members including top-up funding by Austria, Belgium, France, Greece, Italy, Latvia, Netherlands, and Norway. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union, the Chips Joint Undertaking, or the Italian MUR. Neither the European Union, nor the granting authority, nor Italian MUR can be held responsible for them.

References

[1] W. Xiang, K. Yu, F. Han, L. Fang, D. He, Q.-L. Han, Advanced manufacturing in industry 5.0: A survey of key enabling technologies and future trends, IEEE Transactions on Industrial Informatics 20 (2024) 1055–1068.

[2] J. Zhou, J. Yang, S. Xiang, Y. Qin, Remaining useful life prediction methodologies with health indicator dependence for rotating machinery: A comprehensive review, IEEE Transactions on Instrumentation and Measurement 74 (2025) 1–19.

[3] R. A. Shah, O. Urmonov, H. Kim, Two-stage coarse-to-fine image anomaly segmentation and detection model, Image and Vision Computing 139 (2023) 104817.

[4] Y. Han, L. Han, X. Shi, J. Li, X. Huang, X. Hu, C. Chu, Z. Geng, Novel cnn-based transformer integrating boruta algorithm for production prediction modeling and energy saving of industrial processes, Expert Systems with Applications 255 (2024) 124447.

[5] T. Hu, J. Zhang, R. Yi, Y. Du, X. Chen, L. Liu, Y. Wang, C. Wang, Anomalydiffusion: Few-shot anomaly image generation with diffusion model, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2024.

[6] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.

[7] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[8] X. Xie, M. Mirmehdi, Texems: Texture exemplars for defect detection on random textured surfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 1454–1464.

[9] A. Baitieva, D. Hurych, V. Besnier, O. Bernard, Supervised anomaly detection for complex industrial images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 17754–17762.

[10] G. Gui, B.-B. Gao, J. Liu, C. Wang, Y. Wu, Few-shot anomaly-driven generation for anomaly classification and segmentation, in: Proceedings of the IEEE/CVF European Conference on Computer Vision, 2024, pp. 210–226.

[11] J. Im, Y. Son, J. H. Hong, Fun-ad: Fully unsupervised learning for anomaly detection with noisy training data, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2025, pp. 9429–9438.

[12] J. Lagos, H. Ali, A. Faroque, E. Rahtu, Heterogeneous datasets for unsupervised image anomaly detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2025, pp. 7266–7276.

[13] J. Im, Y. Son, J. H. Hong, Fun-ad: Fully unsupervised learning for anomaly detection with noisy training data, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2025, pp. 9429–9438.

[14] H. Kashiani, N. A. Talemi, F. Afghah, Roads: Robust prompt-driven multi-class anomaly detection under domain shift, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2025, pp. 7897–7906.

[15] Z. You, L. Cui, Y. Shen, K. Yang, X. Lu, Y. Zheng, X. Le, A unified model for multi-class anomaly detection, in: Advances in Neural Information Processing Systems, 2022, pp. 4571–4584.

[16] W. Ma, Q. Yao, X. Zhang, Z. Huang, Z. Jiang, S. Zhou, Towards accurate unified anomaly segmentation, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2025, pp. 1342–1352.

[17] X. Li, X. Tan, Z. Chen, Z. Zhang, R. Zhang, R. Guo, G. Jiang, Y. Chen, Y. Qu, L. Ma, et al., One-for-more: Continual diffusion model for anomaly detection, arXiv preprint arXiv:2502.19848 (2025).

[18] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115.

[19] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.

[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: 2017 IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[21] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, X. Hu, Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks , in: Proceedings of the IEEE/CVF Conference

on Computer Vision and Pattern Recognition Workshops, IEEE Computer Society, Los Alamitos, CA, USA, 2020, pp. 111–119.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021.

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[24] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jegou, Training data-efficient image transformers & distillation through attention, in: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 10347–10357.

[25] J. Zhang, M. Suganuma, T. Okatani, Contextual affinity distillation for image anomaly detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 148–157.

[26] Z. Wen, J. Liu, H. Zhao, Q. Wang, A triple semantic-aware knowledge distillation network for industrial defect detection, Computers in Industry 166 (2025) 104252.

[27] G. E. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, ArXiv abs/1503.02531 (2015).

[28] Z. Hao, J. Guo, K. Han, Y. Tang, H. Hu, Y. Wang, C. Xu, One-for-all: bridge the gap between heterogeneous architectures in knowledge distillation, in: Proceedings of the 37th International Conference on Neural Information Processing Systems, 2023.

[29] T. D. Tien, A. T. Nguyen, N. H. Tran, T. D. Huy, S. T. Duong, C. D. T. Nguyen, S. Q. H. Truong, Revisiting reverse distillation for anomaly detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24511–24520.

[30] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K.-R. Müller, A unifying review of deep and shallow anomaly detection, Proceedings of the IEEE 109 (5) (2021) 756–795.

[31] Y. Fang, Y. Fang, R. Chen, H. Xu, X. Ding, Y. Huang, Demeaned sparse: Efficient anomaly detection by residual estimate, in: Forty-second International Conference on Machine Learning, 2025.

[32] W. Luo, Y. Cao, H. Yao, X. Zhang, J. Lou, Y. Cheng, W. Shen, W. Yu, Exploring intrinsic normal prototypes within a single image for universal anomaly detection, in: Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference, 2025, pp. 9974–9983.

[33] H. Guo, L. Ren, J. Fu, Y. Wang, Z. Zhang, C. Lan, H. Wang, X. Hou, Template-guided hierarchical feature restoration for anomaly detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 6447–6458.

[34] T. Defard, A. Setkov, A. Loesch, R. Audigier, Padim: A patch distribution modeling framework for anomaly detection and localization, in: Pattern Recognition. ICPR International Workshops and Challenges, 2021, pp. 475–489.

[35] J. Yoon, K. Sohn, C.-L. Li, S. O. Arik, T. Pfister, SPADE: Semi-supervised anomaly detection under distribution mismatch, Transactions on Machine Learning Research (2023).

[36] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, P. Gehler, Towards total recall in industrial anomaly detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14318–14328.

[37] J. Hyun, S. Kim, G. Jeon, S. H. Kim, K. Bae, B. J. Kang, Reconpatch: Contrastive patch representation learning for industrial anomaly detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 2052–2061.

[38] M. Lee, J. Choi, Text-guided variational image generation for industrial anomaly detection and segmentation, in: Proceedings of the IEEE/CVF

Conference on Computer Vision and Pattern Recognition, 2024, pp. 26519–26528.

[39] Q. Zhou, G. Pang, Y. Tian, S. He, J. Chen, Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection, in: The Twelfth International Conference on Learning Representations, 2023.

[40] Z. Gu, B. Zhu, G. Zhu, Y. Chen, M. Tang, J. Wang, Anomalygpt: Detecting industrial anomalies using large vision-language models, Proceedings of the AAAI Conference on Artificial Intelligence 38 (3) (2024) 1932–1940.

[41] S. Li, J. Cao, P. Ye, Y. Ding, C. Tu, T. Chen, Clipsam: Clip and sam collaboration for zero-shot anomaly segmentation, Neurocomputing 618 (2025) 129122.

[42] J. Zhu, S. Cai, F. Deng, B. C. Ooi, J. Wu, Do llms understand visual anomalies? uncovering llm's capabilities in zero-shot anomaly detection, in: Proceedings of the 32nd ACM International Conference on Multimedia, MM '24, 2024, p. 48–57.

[43] Z. Yang, Z. Li, A. Zeng, Z. Li, C. Yuan, Y. Li, Vitkd: Feature-based knowledge distillation for vision transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2024, pp. 1379–1388.

[44] K. Wu, J. Zhang, H. Peng, M. Liu, B. Xiao, J. Fu, L. Yuan, Tinyvit: Fast pretraining distillation for small vision transformers, in: Proceedings of the IEEE/CVF European Conference on Computer Vision, 2022, pp. 68–85.

[45] M. Kang, S. Son, D. Kim, Adaptive class token knowledge distillation for efficient vision transformer, Knowledge-Based Systems 304 (2024) 112531.

[46] X. Zheng, Y. Luo, P. Zhou, L. Wang, Distilling efficient vision transformers from cnns for semantic segmentation, Pattern Recognition 158 (2025) 111029.

[47] S. Lin, C. Wang, Y. Zheng, C. Tao, X. Dai, Y. Li, Distill vision transformers to cnns via teacher collaboration, in: Proceedings of the IEEE

International Conference on Acoustics, Speech and Signal Processing, 2024, pp. 5925–5929.

[48] B. Zhao, R. Song, J. Liang, Cumulative spatial knowledge distillation for vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 6146–6155.

[49] K. Batzner, L. Heckler, R. König, Efficientad: Accurate visual anomaly detection at millisecond-level latencies, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 128–138.

[50] Y. Feng, W. Chen, Y. Li, B. Chen, Y. Wang, Z. Zhao, H. Liu, M. Zhou, OmiAD: One-step adaptive masked diffusion model for multi-class anomaly detection via adversarial distillation, in: Forty-second International Conference on Machine Learning, 2025.

[51] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, Y. Wei, Layercam: Exploring hierarchical class activation maps for localization, IEEE Transactions on Image Processing 30 (2021) 5875–5888.

[52] S. Abnar, W. H. Zuidema, Quantifying attention flow in transformers, in: D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4190–4197.

[53] H. Chefer, S. Gur, L. Wolf, Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers , in: 2021 IEEE/CVF International Conference on Computer Vision, 2021, pp. 387–396.

[54] L. Yu, W. Xiang, X-Pruner: eXplainable Pruning for Vision Transformers , in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24355–24363.

[55] L. Yu, W. Xiang, J. Fang, Y.-P. P. Chen, L. Chi, ex-vit: A novel explainable vision transformer for weakly supervised semantic segmentation, Pattern Recognition 142 (2023) 109666.

[56] R. Li, Z. Mai, Z. Zhang, J. Jang, S. Sanner, Transcam: Transformer attention-based cam refinement for weakly supervised semantic segmentation, Journal of Visual Communication and Image Representation 92 (2023) 103800.

[57] Z. Peng, Z. Guo, W. Huang, Y. Wang, L. Xie, J. Jiao, Q. Tian, Q. Ye, Conformer: Local features coupling global representations for recognition and detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2023) 9454–9468.

[58] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jegou, M. Douze, LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference , in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12239–12249.

[59] J. Zhang, H. Peng, K. Wu, M. Liu, B. Xiao, J. Fu, L. Yuan, Minivit: Compressing vision transformers with weight multiplexing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12135–12144.

[60] S. Zhang, J. Liu, Feature-constrained and attention-conditioned distillation learning for visual anomaly detection, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2024, pp. 2945–2949.

[61] S. Damm, M. Laszkiewicz, J. Lederer, A. Fischer, Anomalydino: Boosting patch-based few-shot anomaly detection with dinov2, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2025, pp. 1319–1329.

[62] V. Zavrtanik, M. Kristan, D. Skočaj, Draem - a discriminatively trained reconstruction embedding for surface anomaly detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8330–8339.

[63] Y. Li, A. Goodge, F. Liu, C.-S. Foo, Promptad: Zero-shot anomaly detection using text prompts, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 1093–1102.

[64] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever,

Learning transferable visual models from natural language supervision, in: Proceedings of the International Conference on Machine Learning, 2021, pp. 8748–8763.

[65] Q. Fang, Q. Su, W. Lv, W. Xu, J. Yu, Boosting fine-grained visual anomaly detection with coarse-knowledge-aware adversarial learning, Proceedings of the AAAI Conference on Artificial Intelligence 39 (2025) 16532–16540.

[66] H. Mirzaei, M. Nafez, J. Habibi, M. Sabokrou, M. H. Rohban, Adversarially robust anomaly detection through spurious negative pair mitigation, in: The Thirteenth International Conference on Learning Representations, 2025.

[67] Y. Duan, Y. Hong, L. Niu, L. Zhang, Few-shot defect image generation via defect-aware feature manipulation, in: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'23/IAAI'23/EAAI'23, 2023.

[68] J. Wyatt, A. Leach, S. M. Schmon, C. G. Willcocks, Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2022, pp. 649–655.

[69] V. Livernoche, V. Jain, Y. Hezaveh, S. Ravanbakhsh, On diffusion modeling for anomaly detection, in: The Twelfth International Conference on Learning Representations, 2024.

[70] Z. Dai, S. Zeng, H. Liu, X. Li, F. Xue, Y. Zhou, Seas: Few-shot industrial anomaly image generation with separation and sharing fine-tuning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025.

[71] H. Sun, Y. Cao, H. Dong, O. Fink, Unseen visual anomaly generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2025, pp. 25508–25517.

[72] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.

[73] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, p. 6000–6010.

[74] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (2020) 318–327.

[75] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, C. Steger, The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection, International Journal of Computer Vision 129 (4) (2021) 1038–1059.

[76] Y. Cheng, Y. Cao, R. Chen, W. Shen, Rad: A comprehensive dataset for benchmarking the robustness of image anomaly detection, in: 2024 IEEE 20th International Conference on Automation Science and Engineering, 2024, pp. 2123–2128.

[77] Y. Huang, C. Qiu, Y. Guo, X. Wang, K. Yuan, Surface defect saliency of magnetic tile, in: 2018 IEEE 14th International Conference on Automation Science and Engineering, 2018, pp. 612–617.

[78] R. Ding, L. Dai, G. Li, H. Liu, Tdd-net: a tiny defect detection network for printed circuit boards, CAAI Transactions on Intelligence Technology 4 (2) (2019) 110–116.

[79] A. Telea, An image inpainting technique based on the fast marching method, Journal of Graphics Tools 9 (1) (2004) 23–34.

[80] A. Chattopadhay, A. Sarkar, P. Howlader, V. N. Balasubramanian, Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2018, pp. 839–847.

[81] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, P. Dollár, Designing network design spaces, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10425–10433.

[82] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, 2015.