GeneralAD: Anomaly Detection Across Domains by Attending to Distorted Features

Luc P.J. Sträter*, Mohammadreza Salehi*, Efstratios Gavves, Cees G. M. Snoek, and Yuki M. Asano

University of Amsterdam, The Netherlands
lucstrater@gmail.com, s.salehidehnavi@uva.nl

Abstract. In the domain of anomaly detection, methods often excel in either high-level semantic or low-level industrial benchmarks, rarely achieving cross-domain proficiency. Semantic anomalies are novelties that differ in meaning from the training set, like unseen objects in self-driving cars. In contrast, industrial anomalies are subtle defects that preserve semantic meaning, such as cracks in airplane components. In this paper, we present GeneralAD, an anomaly detection framework designed to operate in semantic, near-distribution, and industrial settings with minimal per-task adjustments. In our approach, we capitalize on the inherent design of Vision Transformers, which are trained on image patches, thereby ensuring that the last hidden states retain a patch-based structure. We propose a novel self-supervised anomaly generation module that employs straightforward operations like noise addition and shuffling to patch features to construct pseudo-abnormal samples. These features are fed to an attention-based discriminator, which is trained to score every patch in the image. With this, our method can both accurately identify anomalies at the image level and also generate interpretable anomaly maps. We extensively evaluated our approach on ten datasets, achieving state-of-the-art results in six and on-par performance in the remaining for both localization and detection tasks. Code available at https://github.com/LucStrater/GeneralAD.

Keywords: Anomaly Detection · Self-Supervised Learning · Anomaly Localization

1 Introduction

In Anomaly Detection (AD), the task is to learn the distribution of a given training dataset and distinguish any test sample that does not belong to it. Sample applications vary widely, including the detection of anomalous objects in self-driving cars, which typically operate at a semantic level, and the identification of defects in industrial assembly lines, where the focus is more on low-level elements like pixels [40]. During training, all the samples are labeled as "normal",

* These authors contributed equally to this work.

and there is no access to "abnormal" inputs, which can be formulated similarly to one-class learning tasks. Given this unusual setting of having no access to a complete category that is relevant for testing, this field has spawned many approaches. For example, [11,15,33,38,42] have employed the benefits of pretrained features and diffusion models to learn normal representations that are capable of better solving semantic tasks. Orthogonal to this, [1, 19, 30, 49, 50] have proposed specific architectures to tackle industrial benchmarks for both detection and localization of anomalies.

However, despite the encouraging progress toward better modeling of normal distributions, a growing chasm has been created between the methods that perform well on semantic benchmarks vs on industrial ones. For instance, leading models in semantic benchmarks, such as Transformaly [11] and MSAD [38], demonstrate a significant drop in performance when applied to industrial benchmarks like MVTec-AD [4]. This trend is mirrored in the context of industrial-focused models like SimpleNet [30] and Recontrast [21], which show similar underperformance on semantic datasets, including CIFAR-10 [27]. Despite both sets of methods often utilizing the same strong pretrained features, their one-sided metiers areas suggest overfitting of methodology for specific datasets.

In this work, we aim to produce an all-rounder model that performs well across different tasks and datasets with minimum per-task modification, towards General Anomaly Detection (GeneralAD). Starting with a pretrained Vision Transformer [34] feature extractor, first, we introduce a self-supervised anomaly feature generation module, which gets features of normal samples as the input and generates abnormal ones by applying simple operations such as adding noise and shuffling patches. This results in the generation of high-quality pseudo-abnormal samples, which are not easily identifiable due to the selection of the small noise magnitudes. Moreover, the shuffling introduces logical anomalies, further complicating their detection. Second, we propose to use a transformer-based discriminator, which takes patch features as the input and is trained to detect structural and logical anomalies by attending to different features at different locations. This creates a versatile anomaly detection discriminator, capable of identifying anomalies at varying levels as required, ranging from patch-level to image-level anomalies. Finally, our method is not only able to detect anomalies at the image level but also produce interpretable anomaly maps that can be used to pinpoint abnormal patches for both semantic and industrial tasks.

We evaluate the model across ten different datasets from different benchmarks such as CIFAR-10 [27], CIFAR-100 [27], Fashion-MNIST [45] and View [24] for semantic anomaly detection, Aircraft-FGVC [32] and Stanford-Cars [26] for near anomaly detection [33], where anomaly distribution is very close to normal one, and MVTec-AD [4], MVTec-LOCO [3], VisA [52], and MPDD [25] for industrial anomaly detection. Our results show that the proposed method matches state-of-the-art performance on datasets like FMNIST [45], Stanford-Cars [26], MVTec-AD [4], and VisA [52], and surpasses it in both detection and localization on all other datasets. Overall, this paper makes the following contributions:

- We introduce a self-supervised anomaly feature generation module that mimics a wide range of anomalies, from pixel-level to semantic, by applying a simple and diverse set of operations in the feature space.
- We propose a transformer-based discriminator that effectively operates from individual patches to subsets of patches and entire images. This enables the discriminator to identify and pinpoint a wide range of anomalies in the input data, including structural, logical, and semantic inconsistencies.
- We evaluate our method by conducting comprehensive experiments on three different benchmarks and ten datasets, achieving state-of-the-art results in six out of ten cases. This shows the generality and applicability of the model for different tasks with a minimum amount of per-task modification.

2 Related Work

Anomaly detection encompasses various types of irregularities, each traditionally requiring specific detection methods. In this section we show the distinct areas that intersect in our work and discuss related methods.

2.1 Semantic Anomaly Detection.

Methods in the semantic setting are designed to identify novelties that deviate semantically from the training data distribution. When the deviation between the normal and abnormal distributions in the evaluation dataset is large, it is referred to as an anomaly detection task, as seen in the CIFAR-10 dataset. Conversely, if the deviation is small, it is termed near anomaly detection, exemplified by the Aircraft-FGVC dataset [33]. Common solutions in this domain include self-supervised learning methods [18, 22, 38, 44] and leveraging features from pretrained models [11, 33, 37, 42]. Transformaly [11], for instance, extends the student-teacher architecture proposed in KDAD [42] and achieves state-of-the-art results on semantic anomaly detection benchmarks. FYTIMI [33], the state-of-the-art in near semantic anomaly detection, uses diffusion models [23] to generate pseudo-anomalies, which are then used to solve a binary classification task between the normal dataset and the generated abnormal inputs. Although these methods have achieved superior results on semantic datasets, their ability to detect patch-level anomalies (defections) is limited, resulting in poor performance on industrial benchmarks.

2.2 Industrial Anomaly Detection.

Methods in the industrial setting use intact training samples to identify defects during testing. It typically focuses on fine-grained anomalies, where only small parts of an image differ from the norm. This distinguishes it from semantic and near anomaly detection, where anomalies span the entire image. There are two main approaches: synthesizing-based [28, 48] and embedding-based [2, 6, 12, 14, 20, 21, 37, 39, 46], methods. Synthesizing-based methods attempt to approximate

the abnormal distribution by employing augmentations [28] or feature space manipulation to facilitate the detection task [30]. Embedding-based methods, such as Recontrast [21] and PatchCore [39], focus on transforming normal image features into a space where anomalies stand out. While these approaches excel at identifying pixel-level anomalies, they are less effective at detecting semantic ones. For instance, SimpleNet [30], the state-of-the-art model for this benchmark, is designed to detect anomalies only at the level of local patches. Consequently, it lacks a global perspective necessary for modeling cross-correlations between patches, which is essential for effective semantic anomaly detection. GeneralAD, however, is designed to detect anomalies not only at the patch level but also by identifying abnormal correlations between patch features. This capability makes it a versatile solution for various types of anomalies.

2.3 Self-supervised Anomaly Detection

Self-supervised learning approaches have been applied to detect both semantic anomalies [5, 18, 22, 38, 44] and industrial anomalies [28, 30]. However, the way the proxy task is set up differs depending on the type of anomaly being targeted. For semantic anomalies, the proxy task often involves learning high-level features that capture the normal data distribution, such as through rotation prediction [18], perturbation prediction [5] or contrastive learning [44]. On the other hand, for industrial anomalies, the proxy task typically focuses on learning low-level features that are sensitive to local irregularities, such as through patch-based masking [28]. GeneralAD is designed to predict whether pretrained features have been edited, with the set of editing operations tailored to the type of anomaly to be detected.

3 Method

The problem of *Anomaly Detection* is defined using the following setup [8, 9, 29, 35, 40]. Let $\mathcal{P}$ be a joint probability distribution defined over the input space $\mathcal{X}$ and the output space $\mathcal{Y}$. The output space $\mathcal{Y}$ contains only one class label, which is "normal", i.e., $|\mathcal{Y}| = 1$. Let $\mathcal{D}_{\text{Normal}}$ denote the marginal distribution of $\mathcal{P}$ over $\mathcal{X}$. A neural network $\mathcal{F}$ is trained on samples drawn from $\mathcal{D}_{\text{Normal}}$ to generate a feature embedding, which is subsequently used to compute the anomaly score of an input sample. The objective of anomaly detection is to construct a decision function $\mathcal{M}$ such that for any given test input $x \in \mathcal{X}$ Equation 1 holds.

$$\mathcal{M}(x, \mathcal{F}) = \begin{cases} 0 & \text{if } x \sim \mathcal{D}_{\text{Normal}}, \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

Our method aims to make a framework in which, the decision function $\mathcal{M}$ is trained without relying on specific designs for particular datasets or tasks. To do so, we assume current state-of-the-art pretrained networks [7, 34] can be used to extract semantic feature representations as done by $\mathcal{F}$. Now, to train $\mathcal{M}$, we

need access to both normal and abnormal samples, which is not the case in our setup. To solve the issue, we attempt to approximate them by proposing the self-supervised anomaly feature generation module. This module creates abnormal features by making structural or logical anomalies in the feature space. Finally, to find a decision function $\mathcal{M}$ with minimal design biases, positional embeddings are added to the features, and they are passed to a cross-patch attention discriminator module, which employs a multi-head attention mechanism to detect abnormal regions. During inference time, each input is first passed to the feature extractor, then the extracted patch features are forwarded to the discriminator to get per-patch anomaly scores. For image-level scores, the average of a subset of the patches is considered as the abnormality score. Our method is visualized in Figure 1 and explained in more detail in the following parts.

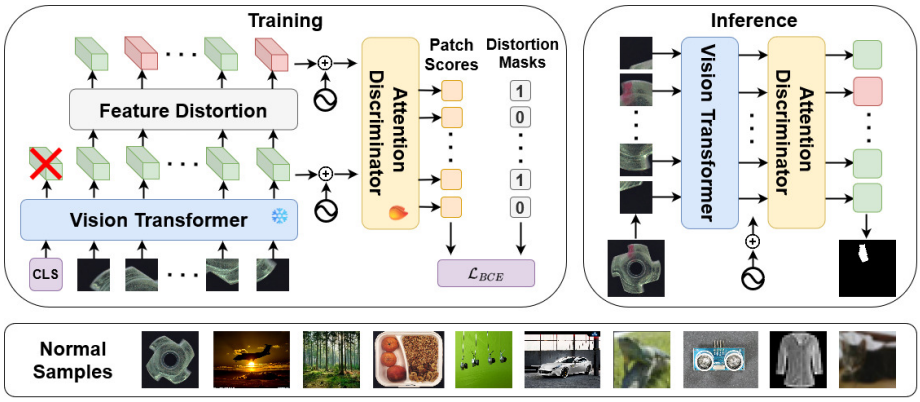

Fig. 1: The proposed method overview. During the training, first, an image is segmented into smaller patches and passed to the pretrained vision encoder to extract patch features. Next, the extracted patches are distorted by the self-supervised anomaly feature generation module, labeled as feature distortion. Finally, all the patches are passed to the cross-patch attention discriminator, whose role is to detect semantic, logical, or structural distortions. During the inference, the feature distortion module is deactivated, and the discriminator is used for both detection and localization tasks. The lower part of the figure displays various normal distributions.

3.1 Feature Extraction

Consider an image represented as $x \in \mathbb{R}^{3\times H\times W}$, which is decomposed into patches of size $P\times P$, resulting in $N{=}\frac{H}{P}\times\frac{W}{P}$ patches, labeled as $x_j^p, j \in 1,\ldots,N$. These patches are then processed through a pre-trained encoder $\mathcal{F}$, generating a set of spatial tokens and one classification token. As the classification token is specifically tailored for semantic tasks, and our method aims to perform well across various tasks, we will exclusively utilize spatial tokens expressed as

$\mathcal{F}(x)$=$[f_1, \ldots, f_N]$. In addition to spatial tokens, we extract their corresponding attention maps for each head i, $\mathcal{A}_i(x) = [a_1, \ldots, a_N]$, to better utilize more informative regions. Here, the attention value for each patch is with respect to the classification token. For subsequent stages in our pipeline, the goal is to learn a decision function $\mathcal{M}$ defined over $\mathcal{F}(x)$, which generalizes well to unseen normal test time inputs yet can not do the same for abnormal ones.

3.2 Self-supervised Anomaly Feature Generation

Having extracted pretrained features $\mathcal{F}(x)$ and the attention maps $\mathcal{A}_i(x)$, we know that patch representations are semantic [7,41,51]; therefore, they can generalize to unseen normal inputs well. Yet, refining generalization boundaries often necessitates access to abnormal distributions, a step unfeasible in our context. Existing studies, such as [33], try to create these distributions through the outputs of early-stage trained diffusion models. However, this approach is costly and fails to address structural or logical anomalies usually found in industrial applications. To address this limitation, we propose a self-supervised anomaly feature generation module (SAG), which receives $\mathcal{F}(x)$ and $\mathcal{A}_i(x)$ as the input and generates anomalies in the feature space by adding noise to random locations or copy-pasting features to strongly attended regions. The resulting *distorted* features are called $\mathcal{F}\epsilon(x)$, and the map of distorted locations is called the distortion mask. The distortion parameters are adjusted by ϵ in $\mathcal{F}\epsilon(x)$:

$$\mathcal{F}\epsilon(x)=\text{SAG}(\mathcal{F}(x), \mathcal{A}_i(x)). \tag{2}$$

We explore three distinct distortion strategies to obtain $\mathcal{F}\epsilon(x)$: (1) *Noise All Patches*, (2) *Noise Random Patches*, and (3) *Attention Shuffle*. The first strategy involves adding Gaussian noise to the features of all patches in the input, while the second strategy applies Gaussian noise to a randomly selected fraction of patches. In the third strategy, important patches are selected by uniformly sampling an attention head from the backbone and identifying N patches with the highest attention values. Here, the attention value for each patch is taken with respect to the classification token and N is uniformly sampled between 1 and the total number of patches. These patches are then shuffled in the feature space.

3.3 Cross-Patch Attention Discriminator

Upon generating anomalous feature maps, denoted as $\mathcal{F}_\epsilon(x)$, and their normal counterparts $\mathcal{F}(x)$, our objective is to effectively train a discriminator. This discriminator, when presented with an input I—which could be either $\mathcal{F}_\epsilon(x)$ or $\mathcal{F}(x)$—should be capable of identifying anomalous patches. In pursuit of ensuring robust performance across a diverse range of anomaly types, it is imperative for the discriminator to discern semantic, structural, or logical irregularities present in $\mathcal{F}_\epsilon(x)$. To achieve this, we propose an approach that initially employs a Multi-Head Attention (MHA) mechanism. This mechanism is designed

to combine information from all patch features. Subsequently, these fused patch features are processed through a Multi-Layer Perceptron (MLP) network, which is responsible for calculating the anomaly score for each individual patch i shown by $P_{\text{score}}(i)$. Finally, positional embeddings E_{pos} are added to the discriminator's input to give it a sense of the input feature positions:

$$F_{\text{fused}}=\text{MHA}(I + E_{\text{pos}}), \tag{3}$$

$$P_{\text{score}}=\text{MLP}(\text{LN}(F_{\text{fused}})). \tag{4}$$

The training objective function is defined as the cross entropy loss between the patch scores and distortion masks.

3.4 Inference

During testing, we derive an image-level anomaly score from patch-level scores by picking the top K highest patch scores and averaging them. This choice of K depends on the size of the anomaly region in the dataset.

$$I_{\text{score}} = \frac{1}{K} \sum_{i=1}^{\text{Top } K} P_{\text{score}}(i). \tag{5}$$

4 Experiments

In the subsequent sub-sections, we present the datasets, implementation details, results, and ablation study.

4.1 Datasets

Our experiments encompass three distinct benchmarks. First, we evaluate our approach on four *semantic anomaly detection* datasets: CIFAR-10 [27], CIFAR-100 [27], Fashion-MNIST [45], and View [24]. In this setting, the model is trained using a single normal class from the dataset and subsequently tested against the remaining classes, which are treated as anomalies. Anomalies in these datasets typically span the entire image.

Second, we explore *near anomaly detection*, focusing on identifying subtle deviations within datasets [33]. We test on the two main benchmarks proposed for this purpose: Aircraft-FGVC [32] and Stanford Cars [26].

Finally, we experiment on four *industrial anomaly detection* datasets: MVTec-AD [4], MVTec-LOCO [3], VisA [52], and MPDD [25]. In these datasets, anomalies are subtle defects, such as scratches, dents, contaminations, and structural changes. MVTec-LOCO also includes logical anomalies, where objects appear in incorrect locations. Unlike the semantic anomaly datasets, anomalies in industrial datasets only cover small sub-regions of the image, with the majority of the image being normal.

4.2 Training Details

This section provides an overview of the training details of our proposed pipeline.

Feature Extraction. In our method, we use the last layer features, followed by normalization using the layer norm component inherent to DINOv2 [34]. This normalization step is crucial for stabilizing the feature representation, thereby enhancing the model's ability to process and analyze the input data effectively. We omit image augmentations from our preprocessing steps to ensure our method's general applicability. However, to maintain consistency with the ViT architectures and to optimize input representation, we rescale all input images to a resolution of 518×518.

Anomalous Feature Generation. For semantic anomaly detection benchmarks we utilize the distortion strategy *Noise All Patches*. The added noise, ϵ, follows a Gaussian distribution $\mathcal{N}(0, 0.25)$. For the industrial anomaly detection benchmarks, the fake features are generated using *Noise Random Patches*. Lastly, for the MVTec-LOCO dataset, which includes logical anomalies, our method uses *Noise Random Patches* and *Attention Shuffle*.

Discriminator. To train the discriminator, we use the AdamW optimizer [31] with an initial learning rate of 0.0005. The learning rate follows a cosine annealing schedule with a decay factor of 0.2. The multi-head attention module in our architecture has 4 heads. The MLP component comprises 3 layers with a hidden dimension of 2048, with no bias added to the last linear layer. The dropout rate after both the attention module and the MLP is set to 0.1. For the semantic anomaly detection datasets, due to their extensive size, we adopted a training regimen of 20 epochs, conducting evaluations after every 250 images. Conversely, for near anomaly detection and industrial anomaly detection datasets, we extended the training duration to 160 epochs, with evaluations after each epoch.

Inference. In industrial datasets to detect small defective regions, the distinction between normal and abnormal regions may be minimal, thus we set $K = 10$. In contrast, for semantic datasets, we set $K = 1369$, which corresponds to all patches for DINOv2, since anomalies typically span the entire image.

4.3 Main results

In this section, we showcase our comparative results against state-of-the-art techniques. We specifically target Transformaly [11] and MSAD [38], recognized as leading approaches in semantic anomaly detection benchmarks. For near anomaly detection benchmarks, GeneralAD is compared against FITYMI [33], although we do not use external data or diffusion models to augment the training dataset, unlike FITYMI. Additionally, we include comparisons with SimpleNet [30] and Reconstrast [21], which are acknowledged as top-performing

methods in industrial anomaly detection. To ensure a fair comparison, we modified the backbone of KDAD [42] from VGG-16 [43] to a vision transformer and included it as well. We report the AUROC performance for both image-level and pixel-level tasks in tables 1 and 2. We provide the results of our method with three model sizes of the DINOv2 backbone. Our model performs inference at a rate of 154, 52, and 17 frames per second on an NVIDIA A100-SXM4-40GB GPU for the small, base, and large variants, respectively.

Image-Level Detection. We present the results in Table 1. As shown, our method not only performed consistently well across various tasks but also surpassed the current state-of-the-art in most of the datasets within different benchmarks. Particularly, we pass Transformaly [11] by $\sim$ 1% on semantic anomaly detection and by $\sim$6% on near anomaly detection benchmarks on average. Furthermore, we surpassed FITYMI [33] by $\sim$1% on near anomaly detection benchmarks. Our method also performed well on industrial anomaly detection datasets, outperforming SimpleNet [30] and Recontrast [21] on MVTec-LOCO by $\sim$7% and $\sim$3% while performing on-par with them on MVTec-AD.

Table 1: Image-level AUROC scores. We compared our method against the current state-of-the-art, encompassing three distinct benchmarks and ten diverse datasets. Unlike other methods that are optimized for peak performance on specific task sets, our approach consistently demonstrated either superior or comparable results across all benchmarks. This underscores the versatility and generality of our proposed method. We report the performance with three model sizes of the DINOv2 backbone. * shows that the method is upgraded with DINOv2-B backbone and reported by us.

	Anomaly Detection				Near Anomaly Detection		Industrial Anomaly Detection			
Method	C-10	C-100	FMNIST	View	Aircraft	St-Cars	MVAD	MVLOCO	VisA	MPDD
Transformaly	98.3	97.3	94.4	<u>95.8</u>	84.0	86.7	87.9	-	-	-
KDAD*	98.5	97.4	94.4	-	85.8	-	85.8	67.5	85.7	72.1
MSAD	97.2	96.4	94.2	-	79.8	87.1	85.5	-	-	-
PANDA	96.2	94.1	**95.6**	93.6	77.7	<u>87.6</u>	86.5	-	-	-
FITYMI	<u>99.1</u>	<u>98.1</u>	79.9	-	88.7	**90.8**	86.4	-	-	-
RDAD	86.5	-	95.0	-	-	-	98.4	79.7	<u>96.0</u>	92.7
Patchcore	67.2	64.1	77.4	-	67.8	78.3	99.2	80.3	94.2	82.1
SimpleNet	86.5	69.8	87.4	76.8	83.6	81.8	**99.6**	77.6	87.9	94.8
Recontrast	84.1	84.0	92.4	-	-	-	<u>99.5</u>	82.1	**97.5**	-
This paper (small)	97.7	96.6	93.9	<u>95.8</u>	89.5	82.2	98.7	81.9	94.4	96.2
This paper (base)	<u>99.1</u>	98.0	94.6	<u>95.8</u>	<u>93.4</u>	82.9	99.2	<u>84.7</u>	<u>96.0</u>	**98.0**
This paper (large)	**99.3**	**98.4**	<u>95.2</u>	**95.9**	**94.6**	87.3	99.2	**84.9**	95.9	<u>97.8</u>

We attribute this increased performance in the semantic anomaly detection benchmarks compared to previous self-supervised anomaly detection methods like SimpleNet to the differences in the discriminator architecture and training approach. The discriminator takes the features of all the patches of the image as input *simultaneously*, while SimpleNet's discriminator runs inference over every patch *separately*. By including attention among patches, our discriminator

detects *global* semantic shifts, which is particularly evident in (near) anomaly detection tasks. Moreover, for industrial anomalies, having access to all the patches in the discriminator allows for adding noise to only a subset of the patches, mimicking the fact that not all the patches in an anomalous image are necessarily anomalous. Furthermore, our model outperforms other methods on logical anomalies due to the self-supervised anomaly feature generation module, which includes adding noise and shuffling a subset of the patches. This shuffling mimics anomalies where objects or components are in the wrong location.

Pixel-Level Localization. We demonstrate the localization capability of our method in Table 2. While almost all state-of-the-art methods in semantic benchmarks perform poorly on anomaly localization, the proposed method shows strong performance in this task. GeneralAD achieves state-of-the-art pixel-level performance on VisA and performs on par with other methods on MVTec-AD, despite not being specifically designed for such datasets. Furthermore, our localization is not restricted to industrial tasks and can be employed to improve the interpretability of decisions for semantic datasets, as shown in Figure 2 and 3.

Table 2: Pixel-level AUROC scores. Our method performed close to state-of-the-art methods that are specifically designed for industrial benchmarks. As opposed to such methods, we were able to provide interpretable maps for semantic tasks as well, demonstrating that the method is more general and applicable across different tasks.

Method	RDAD	Patchcore	SimpleNet	Recontrast	This paper		
					small	base	large
MVTec-AD	97.8	<u>98.1</u>	<u>98.1</u>	**98.4**	97.0	97.2	97.7
VisA	90.1	<u>98.8</u>	91.8	98.2	98.2	<u>98.8</u>	**99.0**
Mean	94.0	**98.5**	95.0	98.3	97.6	98.0	<u>98.4</u>

In Figure 2, we show the qualitative results of our method across a wide range of anomaly detection tasks. As shown, the method provides interpretable maps for semantic datasets, which can improve safety and reliability when deployed in real-world scenarios, such as self-driving cars. Furthermore, it produces precise anomaly segmentation maps for logical and structural anomalies when evaluated on industrial tasks. This supports the generality of our method, which can be used in a wide range of applications.

In Figure 3, we qualitatively compare GeneralAD against state-of-the-art methods KDAD [42] and SimpleNet [30] on semantic anomaly localization. KDAD, despite outperforming other methods in image-level semantic anomaly detection, generates a high number of false positive pixels in the localization map. SimpleNet, on the other hand, does not consistently focus on the most relevant parts, resulting in lower true positive results, albeit with lower false positive

rates. Our method, however, infers the semantic correlation between different patches and produces localization maps with a high true positive rate and low false positive rate, demonstrating its superiority in anomaly localization.

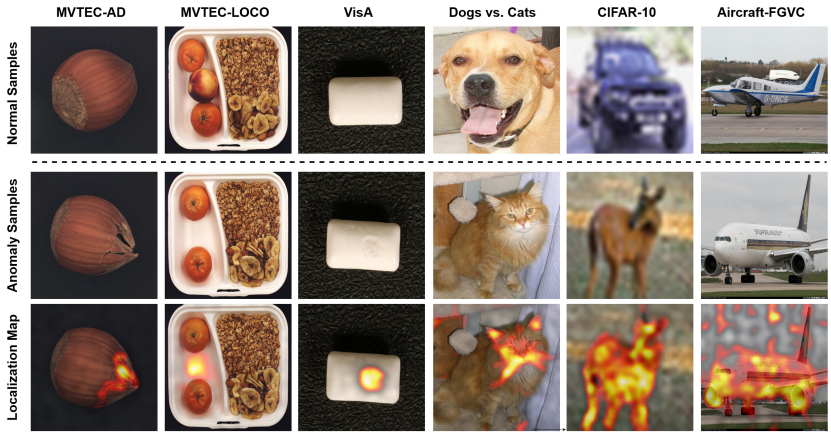

Fig. 2: Qualitative localization results. In the first row, normal samples on which the model is trained are shown. In the second row, we show the real anomaly samples; in the third, we show our localization maps. Our method provides interpretable anomaly segmentation maps for both industrial and semantic tasks. For example, when trained on dog images [17], it can explain why a cat is an abnormal input. Similarly, when the normal class is cars, the entire object from a different class is localized as an anomaly.

Few-shot Anomaly Detection. Few-shot anomaly detection (FSAD) has been introduced to address the needs of quick manufacturing transitions [40]. Despite involving a training stage, our pipeline consistently outperforms methods like PatchCore [39] by large margins. This demonstrates that GeneralAD is also sample-efficient, making it more practical for real-world applications.

Table 3: Few-shot anomaly detection AUROC scores on MVTec AD. Each column indicates the number of training samples used in the experiment. The results are reported over 5 random seeds.

Shots	1	2	4	8
SPADE	71.6	73.4	82.8	84.0
PaDiM	76.1	78.9	80.5	82.0
PatchCore	84.1	87.2	88.5	92.2
This paper (small)	84.4±1.0	86.8±0.4	90.0±1.2	90.5±1.4
This paper (base)	87.0±1.4	**91.9±0.9**	**93.1±0.8**	93.5±0.8
This paper (large)	**87.5±2.2**	91.5±1.4	92.8±1.7	**93.6±0.6**

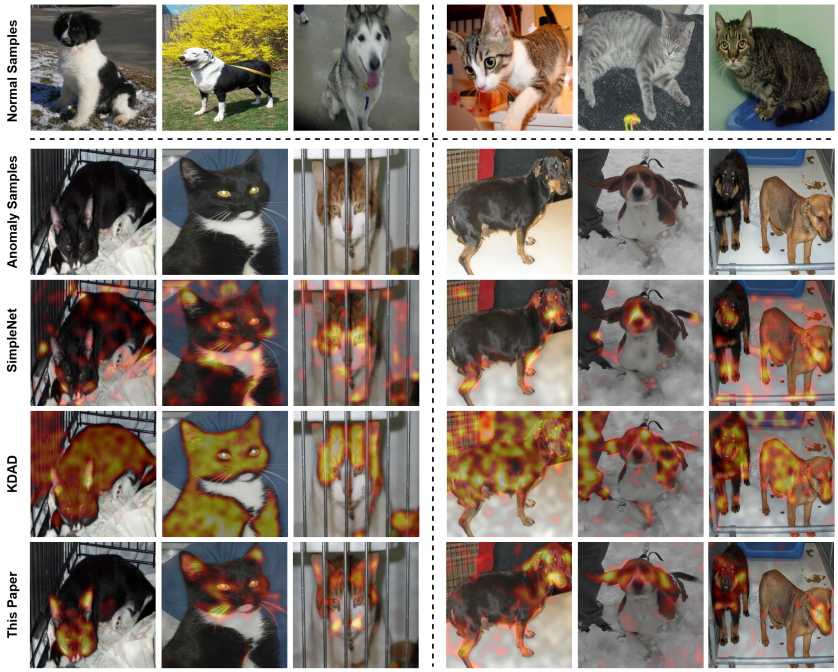

Fig. 3: Comparison of localization maps. We conducted a qualitative comparison between the localization maps of our method, SimpleNet, and KDAD. Qualitatively our method shows higher true positive rate and lower false positive rate, thus providing better semantic localization maps.

4.4 Ablations

Pretraining and Backbone. To demonstrate that our results are not solely due to the use of a state-of-the-art backbone but rather the combination of the backbone and our method, we evaluated our method and other state-of-the-art methods using different backbone architectures [7, 10, 13, 16, 36, 47]. The results are presented in Table 4. Semantic state-of-the-art methods such as KDAD and Transformaly significantly fall behind the state-of-the-art on industrial datasets regardless of the backbone. Similarly, SimpleNet with different backbones underperforms in the semantic benchmarks. This indicates that such methods have been designed to use specific aspects of the pretrained features by optimizing for particular benchmarks. However, our method was designed to leverage the semantic features of DINOv2 [34] specifically and thus perform consistently across different datasets.

Table 4: Effect of pretraining and backbone. We replaced the backbone of different state-of-the-art methods with DINOv2-B. As shown, no other method could excel in all the benchmarks by varying pretraining. GeneralAD, instead, could effectively exploit the features of DINOv2 and worked generally well across different tasks.

Method	Backbone	CIFAR-10	Aircraft-FGVC	MVTec-AD	MVTec-LOCO
Transformaly	ViT-B/16	98.3	84.0	87.9	65.4
	DINOv2-B/14	98.0	82.3	80.9	65.0
KDAD	ViT-B/16	98.1	86.6	80.2	67.5
	DINOv2-B	98.5	85.8	85.8	60.5
SimpleNet	WideResNet50	86.5	83.6	**99.6**	77.6
	ViT-B/16	85.8	83.7	93.3	78.4
	DINOv2-B/14	83.7	88.7	97.7	81.7
This paper	ViT-B/16	93.0	85.4	83.8	68.1
	OpenClip-B/14	93.9	83.7	98.5	81.9
	DINO-B/8	89.5	75.2	98.2	79.7
	DINOv2-B-reg4/14	**99.3**	**93.4**	98.8	83.2
	DINOv2-B/14	99.1	**93.4**	99.2	**84.7**

***K* and Noise Magnitude.** We evaluate the effect of K in the top K selection and the Gaussian noise magnitude (ϵ) in Figure 4. The results indicate that the method performs consistently across a range of Gaussian noise magnitudes, with the optimal magnitude of 0.25 across different anomaly distributions. For the top K parameter, it is essential to set K close to the size of the anomalies in the input. As shown in Figure 4, for (near) anomaly detection, where anomalies span almost the entire image, this leads to $K = 1369$. For industrial anomaly detection, characterized by subtle defects in small parts of the image, this results in $K = 10$.

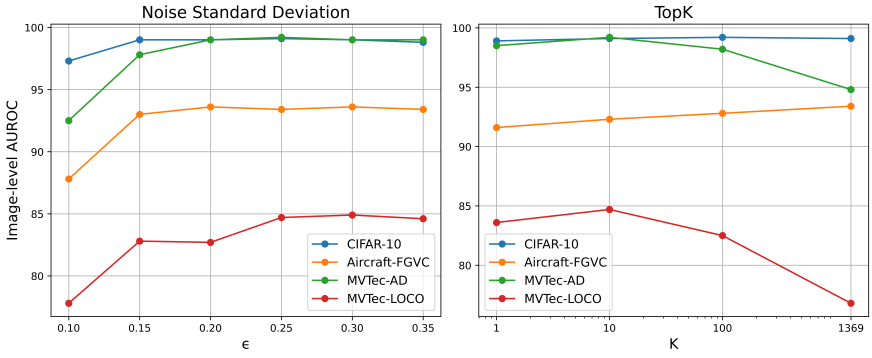

Fig. 4: The effect of K and noise magnitude. Independent of the type of anomaly, the best performance is found with a moderate amount of Gaussian noise. Therefore, we select $\epsilon = 0.25$ for all experiments. The optimal top K parameter depends on the size of the anomalies in the dataset; thus, we choose $K = 1369$ for semantic (near) anomaly detection and $K = 10$ for industrial anomaly detection.

Distortion Types. In Table 5, we show the effect of different feature distortion types across various benchmarks. As shown, each dataset distribution can benefit from specific types of distortions. For semantic datasets, adding noise to all patches gives the best performance, likely due to the distribution of objects where anomalies can cover the entire image. For datasets with anomalies in small regions, such as MVTec-AD, adding noise to random locations performs best. This helps the discriminator identify anomalies within a complex industrial context where normal and abnormal patches coexist.

For datasets with logical anomalies, such as MVTec-LOCO, using more complex distortions, such as shuffling, is beneficial. This technique targets patches with high attention scores, representing the importance of each patch feature, and shuffles them to create anomalies. The model then learns to recognize logical anomalies based on disrupted spatial relations.

Table 5: The effect of distortion types. Different distortion strategies are optimal for different types of anomalies. For semantic datasets, the most effective approach is *Noise All Patches*. In contrast, for industrial datasets that primarily contain structural defects, such as MVTec-AD, *Noise Random Patches* is the most suitable. Finally, for datasets containing logical anomalies, incorporating *Attn Shuffle* proves to be the most effective.

Distortion	CIFAR-10	Aircraft-FGVC	MVTec-AD	MVTec-LOCO
Noise All Patches	**99.3**	**94.6**	99.0	84.1
Noise Random Patches	96.6	94.3	**99.2**	83.8
+ Attention Shuffle	93.6	74.5	99.0	**84.9**

5 Conclusion

In this paper, we have proposed a new approach that significantly narrows the methodological chasm between semantic, near-distribution and industrial benchmarks. Our method leverages a pretrained Vision Transformer (DINOv2) feature extractor alongside a novel self-supervised anomaly feature generation module. This methodology facilitates the creation of pseudo-abnormal samples with subtle, challenging distortions and employs a transformer-based discriminator capable of detecting a wide range of anomalies. It achieves state-of-the-art results in six out of the ten datasets. For the remaining four datasets, our method performs on par with existing standards. This is accomplished without the need for extensive per-task adjustments, in stark contrast to existing works. Finally, our method facilitates the generation of interpretable localization maps, enhancing the understanding and analysis of detected anomalies. *Limitations.* Despite our effort to introduce a generic method that works across all domains, our method does not yet succeed wholly across the board. We believe that the set of diverse benchmarks that we have evaluated in this paper can serve as a springboard for a generation of new and general anomaly detection methods.

Acknowledgements

This research was funded by the University of Amsterdam. The authors acknowledge SURF for providing access to the National Supercomputer Snellius.

References

1. Bae, J., Lee, J.H., Kim, S.: Pni: industrial anomaly detection using position and neighborhood information. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6373–6383 (2023)
2. Batzner, K., Heckler, L., König, R.: Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 128–138 (2024)
3. Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., Steger, C.: Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. International Journal of Computer Vision **130**(4), 947–969 (2022)
4. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9592–9600 (2019)
5. Cai, J., Fan, J.: Perturbation learning based anomaly detection. Advances in Neural Information Processing Systems **35**, 14317–14330 (2022)
6. Cao, T., Zhu, J., Pang, G.: Anomaly detection under distribution shift. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6511–6523 (2023)
7. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
8. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey. arXiv e-prints pp. arXiv–1901 (2019)
9. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 1–58 (2009)
10. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2818–2829 (2023)
11. Cohen, M.J., Avidan, S.: Transformaly-two (feature spaces) are better than one. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4060–4069 (2022)
12. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. arXiv e-prints pp. arXiv–2005 (2020)
13. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. In: The Twelfth International Conference on Learning Representations (2024)
14. Defard, T., Setkov, A., Loesch, A., Audigier, R.: Padim: a patch distribution modeling framework for anomaly detection and localization. In: International Conference on Pattern Recognition. pp. 475–489. Springer (2021)
15. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9737–9746 (2022)

16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
17. Elson, J., Douceur, J.R., Howell, J., Saul, J.: Asirra: a captcha that exploits interest-aligned manual image categorization. CCS **7**, 366–374 (2007)
18. Golan, I., El-Yaniv, R.: Deep anomaly detection using geometric transformations. Advances in neural information processing systems **31** (2018)
19. Gu, Z., Liu, L., Chen, X., Yi, R., Zhang, J., Wang, Y., Wang, C., Shu, A., Jiang, G., Ma, L.: Remembering normality: Memory-guided knowledge distillation for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16401–16409 (2023)
20. Gudovskiy, D., Ishizaka, S., Kozuka, K.: Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 98–107 (2022)
21. Guo, J., Jia, L., Zhang, W., Li, H., et al.: Recontrast: Domain-specific anomaly detection via contrastive reconstruction. Advances in Neural Information Processing Systems **36** (2024)
22. Hendrycks, D., Mazeika, M., Kadavath, S., Song, D.: Using self-supervised learning can improve model robustness and uncertainty. Advances in neural information processing systems **32** (2019)
23. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
24. Intel: Intel Image Classification. `https://www.kaggle.com/datasets/puneet6060/intel-image-classification/data` (2019)
25. Jezek, S., Jonak, M., Burget, R., Dvorak, P., Skotak, M.: Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In: 2021 13th International congress on ultra modern telecommunications and control systems and workshops (ICUMT). pp. 66–71. IEEE (2021)
26. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 554–561 (2013)
27. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
28. Li, C.L., Sohn, K., Yoon, J., Pfister, T.: Cutpaste: Self-supervised learning for anomaly detection and localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9664–9674 (2021)
29. Liu, J., Xie, G., Wang, J., Li, S., Wang, C., Zheng, F., Jin, Y.: Deep industrial image anomaly detection: A survey. Machine Intelligence Research **21**(1), 104–135 (2024)
30. Liu, Z., Zhou, Y., Xu, Y., Wang, Z.: Simplenet: A simple network for image anomaly detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20402–20411 (2023)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
32. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv e-prints pp. arXiv–1306 (2013)
33. Mirzaei, H., Salehi, M., Shahabi, S., Gavves, E., Snoek, C.G.M., Sabokrou, M., Rohban, M.H.: Fake it until you make it : Towards accurate near-distribution novelty

detection. In: The Eleventh International Conference on Learning Representations (2023)
34. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. Transactions on Machine Learning Research (2023)
35. Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep learning for anomaly detection: A review. ACM computing surveys (CSUR) **54**(2), 1–38 (2021)
36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
37. Reiss, T., Cohen, N., Bergman, L., Hoshen, Y.: Panda: Adapting pretrained features for anomaly detection and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2806–2814 (2021)
38. Reiss, T., Hoshen, Y.: Mean-shifted contrastive loss for anomaly detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2155–2162 (2023)
39. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14318–14328 (2022)
40. Salehi, M., Mirzaei, H., Hendrycks, D., Li, Y., Rohban, M., Sabokrou, M., et al.: A unified survey on anomaly, novelty, open-set, and out of-distribution detection: Solutions and future challenges. Transactions on Machine Learning Research (2022)
41. Salehi, M., Gavves, E., Snoek, C.G., Asano, Y.M.: Time does tell: Self-supervised time-tuning of dense image representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16536–16547 (2023)
42. Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M.H., Rabiee, H.R.: Multiresolution knowledge distillation for anomaly detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14902–14912 (2021)
43. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations (ICLR 2015). Computational and Biological Learning Society (2015)
44. Tack, J., Mo, S., Jeong, J., Shin, J.: Csi: Novelty detection via contrastive learning on distributionally shifted instances. Advances in neural information processing systems **33**, 11839–11852 (2020)
45. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv e-prints pp. arXiv–1708 (2017)
46. Xie, X., Huang, Y., Ning, W., Wu, D., Li, Z., Yang, H.: Rdad: A reconstructive and discriminative anomaly detection model based on transformer. International Journal of Intelligent Systems **37**(11), 8928–8946 (2022)
47. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: British Machine Vision Conference 2016. British Machine Vision Association (2016)
48. Zavrtanik, V., Kristan, M., Skočaj, D.: Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8330–8339 (2021)
49. Zhang, X., Li, N., Li, J., Dai, T., Jiang, Y., Xia, S.T.: Unsupervised surface anomaly detection with diffusion probabilistic model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6782–6791 (2023)

50. Zhang, X., Li, S., Li, X., Huang, P., Shan, J., Chen, T.: Destseg: Segmentation guided denoising student-teacher for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3914–3923 (2023)
51. Ziegler, A., Asano, Y.M.: Self-supervised learning of object parts for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14502–14511 (2022)
52. Zou, Y., Jeong, J., Pemula, L., Zhang, D., Dabeer, O.: Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In: European Conference on Computer Vision. pp. 392–408. Springer (2022)