A Hierarchical Transformation-Discriminating Generative Model for Few Shot Anomaly Detection

Shelly Sheynin1* Sagie Benaim1* Lior Wolf1,2
1The School of Computer Science, Tel Aviv University
2Facebook AI Research

Abstract

Anomaly detection, the task of identifying unusual samples in data, often relies on a large set of training samples. In this work, we consider the setting of few-shot anomaly detection in images, where only a few images are given at training. We devise a hierarchical generative model that captures the multi-scale patch distribution of each training image. We further enhance the representation of our model by using image transformations and optimize scale-specific patch-discriminators to distinguish between real and fake patches of the image, as well as between different transformations applied to those patches. The anomaly score is obtained by aggregating the patch-based votes of the correct transformation across scales and image regions. We demonstrate the superiority of our method on both the one-shot and few-shot settings, on the datasets of Paris, CIFAR10, MNIST and FashionMNIST as well as in the setting of defect detection on MVTec. In all cases, our method outperforms the recent baseline methods.

1. Introduction

Anomaly detection [5, 1] is the task of detecting unusual samples in data. In the typical setting of one class classification [20], given a large collection of samples from the normal (non-anomalous) data, the learner is asked to classify novel samples as either normal or anomalous. In this paper, we aim to solve this problem given very few training samples, including the case of a single training sample. Our study is motivated by the scarcity of training samples in many visual domains, as well as by the human ability to solve this task after observing a very limited number of samples [7, 8].

Our model relies on two main components. The first component is a hierarchical generative model that captures the internal patch statistics of one or few images at multiple scales. This component follows the recent successes of deep generative models to generate multiple, varied, natural looking images, given a single training image [28, 12, 35]. We generalize this setting to few images, by adding conditioning on the image index, thus enabling information sharing between multiple training images.

Modeling patches at different scales, allows the detection of anomalies both in global properties, such as color or large-scale structural changes, and in local regions. The multi-scale patch based approach is also useful when considering the task of defect detection, where the anomaly may only manifest in few well localized regions.

The second component is a self-supervised learning task. Recent approaches [10, 3, 11] have shown that, in the many-shot case, a classifier trained on a proxy task, such as identifying the transformation applied to the input, accurately captures novel samples that are similar to the training data, and thus can distinguish between normal and anomalous samples. We utilize such a proxy task in the context of our multi-scale generative model. For each scale, the generated patches are transformed according to a predefined set of transformations. A discriminator is asked to distinguish between real and fake samples, as well as between the transformations performed.

Our method builds these components into a single model. At test time, a sample is considered anomalous, if many of its patches, at all scales, are determined anomalous, according to the learned patch-specific discriminators. In a comprehensive set of the experiments, we demonstrate that the proposed method outperforms recent baselines on anomaly detection benchmarks. This is true for the one-shot, five-shot and ten-shot settings. When the number of training samples increases, our method is shown to improve in performance. In the field of defect detection, we show that our method outperforms, in the few shot setting, the state of the art in localizing the anomalous patch.

To conclude, our work comprises the following contributions: (1) Presenting a method for few-shot anomaly detection which on the conceptual level is novel. (2) The method can easily be extended to the problem of few-shot defect detection. (3) We offer a number of unique components: (i) Multiclass discriminator which is trained adversarially using
labels from SSL task (ii) A novel use of a discriminator at test time (iii) Applying transformations on patches at different scales, and not on full-size images (as [10]). (iv) Extending the hierarchical GAN of [28] to multiple images using a novel conditional generator. (4) The proposed method is extensively evaluated on five datasets and achieves better performance compared to the baselines.

2. Related Work

**Image based anomaly detection** Image based anomaly detection methods can be divided into the categories of reconstruction, classification and distribution based methods [30].

Reconstruction based methods represent data in a manner whereby normal data can be reconstructed with small error, while anomalous data incurs a high reconstruction error.

Classification based methods attempt to discriminate between data regions of normal data and those of anomalous data [31, 39] Recent self-supervised approaches attempt to find a "proxy" classification objective, such that classifying normal data based on this objective, allows for a good separation of normal and anomalous data. For instance, Goyal et al. [11] trains a classifier to distinguish training samples from their perturbations generated adversarially. Golan and El-Yaniv [10] use a set of predefined transformations. They train a classifier to distinguish between the type of transformation performed on normal data and show that such a classifier can be used to distinguish between normal and anomalous data. While our method uses such a proxy objective, it is used within an hierarchical patch based generative setting. This allows the modeling of the patch based distribution of images together with a discriminative capability to model the regions of this distribution.

Distribution based methods model the distribution of normal data. Deep generative models have shown great promise in modeling complicated distributions. In particular, autoencoders and variational autoencoders [40, 2], as well as GAN based approaches [33] are typically used. Nalisnick et al., [21] however, argue that a generative model alone may not be sufficient to detect out-of-distribution inputs.

Unlike these approaches, our model benefits from the use of both distribution and classification based components: (1) A hierarchical generative model used to model the internal multi-scale patch distribution of a single or few images and (2) A scale-dependent discriminator, which learns to distinguish between real and fake image patches, as well as between transformations of such patches. Our model incorporates these components into a single model. Unlike previous approaches, this allows for the detection of anomalous samples in the case where, during training, only a single or a few images are given from the normal class.

**Few shot learning** The use of limited supervision for image classification has been studied extensively [37, 6, 36]. Our work relates more closely to anomaly detection works that use limited supervision. Some works [22, 24] consider the setting where a limited number of samples is given from the anomalous classes, but many samples are given from the normal class. Kozerawski et al., [15] use large labelled dataset generated from ImageNet and apply transfer learning. Other works [23, 25] tackle a slightly different problem, where few samples are given from novel anomalous classes. Frikha et al., [9] and Kruspe [17] consider a meta-learning approach where few samples are given from many classes at training. All of these works use additional supervision, not used in our method. Our work assumes uses only a limited number of samples from the normal class.

**Learning the Internal Statistics** Our work is related to recent works that model the internal distribution of images [42, 35, 28, 12]. However, these methods are used for generation, and are designed on a single image only.

3. Method

Let \( \mathcal{X} \) be the set of all natural images and let \( X \subset \mathcal{X} \) be the subset of images in the class that is defined by the training images, i.e., the normal class. Images residing outside \( \mathcal{X} \) are considered anomalous. We are given an i.i.d randomly selected subset of images \( X_k = \{x_1, \ldots, x_k\} \subset \mathcal{X} \). The task is to learn a classifier \( C_{X_k} \) from the training set \( X_k \), such that \( C_{X_k}(x) = 1 \) if \( x \in \mathcal{X} \) and \( C_{X_k}(x) = 0 \) otherwise. We are interested in the case where \( k \) is small.

### 3.1. One Shot Anomaly Detection

We begin with the case of \( k = 1 \), where the training set contains a single sample \( x_1 \), denoted as \( x \) for brevity. While \( x \) alone cannot capture the intra-class variability of the normal class, we argue that the image itself contains
informative structure that can help identify the class. An illustration of our training pipeline is provided in Fig. 1.

**Internal distribution of patches** Our sample space can be significantly enriched by considering different image patches, at different scales. To this end, we model the distribution of the patches of $x$ at different scales, using a similar pipeline to SinGAN [28]. Unlike SinGAN, we use a multi-class discriminator to distinguish between different classes of transformations applied on real and generated samples.

For a given scale $n = 0, 1, 2, \ldots, N$, we denote by $p^n_x$ the distribution of images that have the same patch distribution at this scale as $x$. In other words, $x$ is composed of many patches at a given distribution and we model the distribution $p^n_x$ of images that have the same scale-dependent patch distribution. For each scale $n$, the downsampling factor is $r^{N-n}$ for some $r > 1$. Scale 0 is of the lowest resolution and scale $N$ is of the highest (original) image resolution. A bicubic downsampling is used.

At each scale $n$, our method employs a patch GAN [43, 19] for generating samples in $p^n_x$. Each patch-GAN consists of a fully convolutional generator $G^n_x$ and discriminator $D^n_x$ with a receptive field of size $11 \times 11$. The architecture of $G^n_x$ and $D^n_x$ and training details are outlined below in Sec. 3.4. Let $x^n$ be $x$, down-sampled by a factor of $r^{N-n}$, and $z^n$ be Gaussian noise of the same dimension and shape as $x^n$. At the coarsest scale, $n = 0$, the output of $G^n_x$ is defined as (the overline is used to denote generated images):

$$\overline{x}^0 = G^n_x(z^0)$$

(1)

Moving to finer scales $n > 0$, $G^n_x$ accepts as input both $z^n$ and an upscaled image of $\overline{x}^{n-1}$ to the dimension of $x^n$, denoted by $\uparrow \overline{x}^{n-1}$. A residual generation is then used:

$$\overline{x}^n = G^n_x(z^n + \uparrow \overline{x}^{n-1}) + \uparrow \overline{x}^{n-1}$$

(2)

This way, the network $G^n_x$ adds the missing details of $\uparrow \overline{x}^{n-1}$ that are specific to scale $n$.

**Transformations** To enhance the ability of our model to capture the coarsest scale, $n > 0$, $G^n_x$ accepts as input both $z^n$ and an upscaled image of $\overline{x}^{n-1}$ to the dimension of $x^n$, denoted by $\uparrow \overline{x}^{n-1}$. A residual generation is then used:

$$\overline{x}^n = G^n_x(z^n + \uparrow \overline{x}^{n-1}) + \uparrow \overline{x}^{n-1}$$

(2)

This way, the network $G^n_x$ adds the missing details of $\uparrow \overline{x}^{n-1}$ that are specific to scale $n$.

To this end, we model the distribution of the patches of $x$ at different scales, using a similar pipeline to SinGAN [28]. Unlike SinGAN, we use a multi-class discriminator to distinguish between different classes of transformations applied on real and generated samples.

For a given scale $n = 0, 1, 2, \ldots, N$, we denote by $p^n_x$ the distribution of images that have the same patch distribution at this scale as $x$. In other words, $x$ is composed of many patches at a given distribution and we model the distribution $p^n_x$ of images that have the same scale-dependent patch distribution. For each scale $n$, the downsampling factor is $r^{N-n}$ for some $r > 1$. Scale 0 is of the lowest resolution and scale $N$ is of the highest (original) image resolution. A bicubic downsampling is used.

At each scale $n$, our method employs a patch GAN [43, 19] for generating samples in $p^n_x$. Each patch-GAN consists of a fully convolutional generator $G^n_x$ and discriminator $D^n_x$ with a receptive field of size $11 \times 11$. The architecture of $G^n_x$ and $D^n_x$ and training details are outlined below in Sec. 3.4. Let $x^n$ be $x$, down-sampled by a factor of $r^{N-n}$, and $z^n$ be Gaussian noise of the same dimension and shape as $x^n$. At the coarsest scale, $n = 0$, the output of $G^n_x$ is defined as (the overline is used to denote generated images):

$$\overline{x}^0 = G^n_x(z^0)$$

(1)

Moving to finer scales $n > 0$, $G^n_x$ accepts as input both $z^n$ and an upscaled image of $\overline{x}^{n-1}$ to the dimension of $x^n$, denoted by $\uparrow \overline{x}^{n-1}$. A residual generation is then used:

$$\overline{x}^n = G^n_x(z^n + \uparrow \overline{x}^{n-1}) + \uparrow \overline{x}^{n-1}$$

(2)

This way, the network $G^n_x$ adds the missing details of $\uparrow \overline{x}^{n-1}$ that are specific to scale $n$.

**Transformations** To enhance the ability of our model to represent $p^n_x$ and inspired by recent work [41, 13] on few shot generation, we apply a fixed set of $M$ differentiable transformations, $T_1, \ldots, T_M$, to all real and generated images given as input to $D^n_x$. These transformations are chosen, such that, on one hand, they enrich the sample space captured by the current model, thus enabling it to more faithfully capture $p^n_x$, while on the other hand, they do not produce samples outside the class of $x$.

We use the set of transformations which result from applying the following transformations sequentially: (1) horizontal flip, (2) translation: the image is shifted by a ratio of $0.15$ in $x$ axis, $y$ axis or both, (3) $90^\circ$ rotations $\{R_0, R_90, R_180, R_270\}$, and (4) color transformation of RGB to gray-scale. For grayscale image datasets, we do not use color transformations (4). Due to memory constraints, we use a subset of $M = 54$ (42 for grayscale image datasets) transformations from this group, where the same set of transformations are used for all datasets and experiments. The exact set of transformations is given in the supplementary (Sec. 1). We also consider another subsets of transformations and report their results as well in the supplementary (Sec. 3.1). We avoid undesirable effects at image borders, by first padding the image with reflection. As a pre-processing step, before applying the transformations, we apply histogram equalization to all the training images in RGB color space.

**Training objectives** $T_1, \ldots, T_M$ are used in training $G^n_x$ and $D^n_x$. For each scale, we transform $x^n$ into the set $\{T_1(x^n), \ldots, T_M(x^n)\}$ of $M$ transformed real images. Similarly, each generated sample $\overline{x}^n$ is transformed into $\{T_1(\overline{x}^n), \ldots, T_M(\overline{x}^n)\}$. Additionally, the task of recognizing the underlying transformation enriches the output space of $D^n_x$, in a way that is suitable for anomaly detection.

$D^n_x$ is a fully convolutional Markovian discriminator [43, 19] of the same architecture as $G^n_x$, except that the number of output channels in the last convolutional layer is $M + 1$. $D^n_x$ is trained in a discriminative fashion to classify all the patches of $T_i(x^n)$ as $i$. In addition, it is optimized to classify all the patches of $T_i(\overline{x}^n)$ to the “fake” class 0. $G^n_x$ tries to fool $D^n_x$ by producing $\overline{x}^n$ such that all the patches of $T_i(\overline{x}^n)$ are classified as 0. For each $T_i$, $D^n_x$ produces $M + 1$ maps of size $H \times H$. Applying softmax, along the $M + 1$ dimensions, produces pseudo-probabilities for patch $p$ of the input to belong to one of the $M + 1$ classes. We use the following adversarial loss terms:

$$\mathcal{L}_{adv}(D^n_x) = \sum_{p=1}^{M} \sum_{i=1}^{M} \mathcal{L}_{CE}(D^n_x(T_i(x^n))_p, i)$$

(3)

$$\mathcal{L}_{adv}(D^n_x) = \sum_{p=1}^{M} \sum_{i=1}^{M} \mathcal{L}_{CE}(D^n_x(T_i(\overline{x}^n))_p, 0)$$

(4)

$$\mathcal{L}_{adv}(G^n_x) = \sum_{p=1}^{M} \sum_{i=1}^{M} \mathcal{L}_{CE}(D^n_x(T_i(\overline{x}^n))_p, i)$$

(5)

$$\mathcal{L}_{adv}(D^n_x, G^n_x) = \mathcal{L}_{adv}(G^n_x) - \mathcal{L}_{adv}(D^n_x)$$

(6)

where $D^n_x(T_i(x^n))_p$ (similarly $D^n_x(T_i(\overline{x}^n))_p$) denotes the softmax probability vector of size $M + 1$ for point $p$ in the $H \times H$ map produced by $D^n_x$ and $\mathcal{L}_{CE}$ denotes the cross entropy loss. Recent literature [10] has shown the potential of training a classifier on transformation detection for anomaly detection. We utilize a similar discriminative objective, but in the context of a multi-scale hierarchical generative model. As shown in Sec. 4, the use of this hierarchical generative mode significantly improves results.

In addition to adversarial training, a reconstruction loss is used. For $n = 0$, $G^n_x$ attempts to reconstruct $x_0$ given a fixed random noise $z^n$ while for $n > 0$, $G^n_x$ attempts to reconstruct $x^n$ given an upscaled version of $\overline{x}^{n-1}$ that is obtained,
where the Anomaly score value is, across all patches and scales, and for all transformations, it can be recursively, based on $z^*$, without further randomness:

$$\mathbb{E}_i^0 = G_i^n(z^*)$$

$$\mathbb{E}_i^n = G_i^n(\mathbb{E}_i^{n-1})$$, for $n > 0$

$$L_{\text{recon}_0}(G_{x_i}^n) = ||\mathbb{E}_i^0 - x_i^0||_2$$

$$L_{\text{recon}_n}(G_{x_i}^n) = ||\mathbb{E}_i^n - x_i^n||_2$$, for $n > 0$

The reconstruction loss is used to control $\sigma^n$, the standard deviation of the Gaussian noise used at each scale, $z^n$, which indicates the level of detail required at each scale. In particular, $\sigma^n = || ↑ x_i^{n-1} - x_i^n ||_2$. Without it (using uniform randomness), $G^n_i$ can tint input images. The overall loss used at scale $n$, for some hyperparameter $\alpha > 0$, is then:

$$\min_{D^n_Z} \max_{G^n_X} \sum L_{\text{adv}}(D^n_Z, G^n_X) + \alpha L_{\text{recon}_n}(G^n_X)$$

### Anomaly score

We cast the problem of anomaly detection as determining if the patches of a given test image, $x_{\text{test}}$, and their transformations, are real. That is, for a given test image $x_{\text{test}}$, and for each scale $n$, let $T_i(x_{\text{test}}^n)$ be the result of applying $M$ transformations on $x_{\text{test}}^n$, the downsampled version of $x_{\text{test}}$ at scale $n$. $D^n_Z(T_i(x_{\text{test}}^n))$ can be viewed as a set of $H \times H$ vectors of size $M + 1$, where each vector corresponds to a given patch of the input. By removing the 0'th element (the “fake” class) and applying softmax, the $i$'th element of this vector denotes the probability that this patch is of class $i$ (the $i$'th transformation). Since $D$ is trained on data from the normal class, the higher this value is, across all patches and scales, and for all transformations $T_i$, the higher our confidence in this sample being non-anomalous. Our anomaly score is, therefore, given as:

$$C_{i}(x_{\text{test}}) = \frac{N}{n} \sum_{n=0}^{M} \sum_{i=1}^{P} \sum_{p \in H \times H} [D^n_Z(T_i(x_{\text{test}}^n))]|_i$$

where $D^n_Z$ outputs the last M maps of $D^n_Z$, i.e., it outputs $H \times H$ vectors of size $M$. Without the 0'th “fake” element. $D^n_Z(T_i(x_{\text{test}}^n))|_i$ denotes the softmax probability vector of the $p$'th patch, and the index $i$ is indexing this vector, to provide the pseudo-probability of the patch belonging to class (transformation) $i$. The lower $C_{i}(x_{\text{test}})$, the more anomalous $x_{\text{test}}$ is. Further analysis on the anomaly score is given in the supplementary (Sec. 3.2).

### 3.2. From one shot to few shot anomaly detection

Moving to the few shot setting, we are now equipped with a subset $X_k = \{x_1, \ldots, x_k\}$ of $X$. To maximize our sample space, we would like to capture the inter-class variability offered by each of the $x_i$'s. We, therefore, devise a model that captures the multi-scale patch distribution of each $x_i$. One possibility is to train a model for each $x_i$ separately, and combine the scores of each model. However, this may be computationally expensive and time consuming. Instead, we use a single generative model that is conditioned on $i$.

**Conditional generation**

To condition the generator on each training sample $x_i$, we concatenate a single channel, whose values are $i$ everywhere, to the input $z$ of the generator, obtaining an input, which we denote as $cat(z, i)$.

Let $x$ be a tensor of dimensions $C \times H \times W$ (channels, height and width). We denote by $cat(x, i)$ the tensor of dimensions $(C + 1) \times H \times W$, where the last channel equals $i$ in all $H \times W$ positions. Generalizing our one-image generator, the generator trained on the set $X_k$ is denoted by $G_{X_k}^n$, and is defined as follows:

$$\mathbb{E}_i^0 = G_{X_k}^0(cat(z^0, i))$$

$$\mathbb{E}_i^n = G_{X_k}^n(cat(z^n + ↑ x_i^{n-1}, i)) + ↑ x_i^{n-1}$$

**Training objectives**

Our losses extend those used in Sec. 3.1 to the few shot case. Let $D^n_{X_k}$ be the discriminator at scale $n$ trained on $X_k$. We define $L_{\text{adv}}(D^n_{X_k}, G^n_{X_k})$ to be $L_{\text{adv}}(D^n_{X_k}, G^n_{X_k})$ (Eq. 6) applied with input $x_i^n$ instead of $z^n$. $L_{\text{adv}}(D^n_{X_k}, G^n_{X_k})$ is defined to be the sum over $L_{\text{adv}}(D^n_{X_k}, G^n_{X_k})$ for all $i$. The reconstruction loss for each sample $x_i$, at scale $n$, now becomes:

$$\mathbb{E}_i^0 = G_{X_k}^0(cat(z^*, i))$$

$$\mathbb{E}_i^n = G_{X_k}^n(cat(↑ x_i^{n-1}, i))$$

$$L_{\text{recon}_n}(G^n_{X_k}) = ||\mathbb{E}_i^n - x_i^n||_2$$

The full reconstruction loss $L_{\text{recon}_n}(G^n_{X_k})$ is the sum over $L_{\text{recon}_n}(G^n_{X_k})$ for all $i$. Note that the generator, in the multi-shot case, learns to generate from multiple image distributions $i$. In other words, we have a set of $k$ derived generators per scale $n$, each of which is given by $G_{X_k}^n(cat(\ldots, i))$. Similar to the one-shot case, the overall loss at scale $n$ is:

$$\min_{G_{X_k}^n} \max_{D_{X_k}^n} L_{\text{multi}}(D_{X_k}^n, G_{X_k}^n) + \alpha L_{\text{recon}_n}(G_{X_k}^n)$$

The architecture of $D_{X_k}^n$ is not modified from the one-shot setting. The difference is that now, $D_{X_k}^n$ is trained on more samples $X_k$ as well as generated samples for each sample $i$, and so captures a richer distribution. The anomaly score used in the $k > 1$ case, denoted $C_{X_k}(x_{\text{test}})$, is the same anomaly score of Eq. 12 with $D_{X_k}^n$ instead of $D_{X_k}^n$.

### 3.3. Defect Detection

We also consider the task of “defect detection” [14], which is a localized variant of anomaly detection. In this variant, normal samples are visually similar and anomalous samples contain subtle local changes. Since our method models the multi-scale patch distribution of each training image, it can be readily used to localize areas where abnormal
features occur. For anomalous samples, only a small number of patches are anomalous, and the rest are similar to patches in normal samples. Therefore, instead of averaging over all patches, we average over the 5% of patches with the lowest anomaly score. Specifically, Eq. 12 is modified to:

$$C(x) = \sum_{n=0}^{N} \sum_{i=1}^{M} \sum_{p \in f_n^i(H \times H)} D^n_{X_k}(x_{test}^n)$$

where \(f_n^i(H \times H)\) denotes 5% of patch indices with lowest anomaly score for scale \(n\) and transformation \(i\). The few shot score is defined with \(D^n_{X_k}\) instead of \(D^n_{X_k}\). The effect of using a different percentage is given in Sec. 4.3.

### 3.4. Architecture and training details

The generator \(G^n_{X_k}\) and the discriminator \(D^n_{X_k}\) each consist of five convolutional blocks. A convolutional block consists of: (i) a 3 × 3 convolutional layer and is padded such that it maintains the spatial resolution of the input, (ii) Batch normalization layer and (iii) LeakyReLU activation with a slope of 0.2. The application of five such blocks results in a fixed effective receptive field of \(11 \times 11\) for generator \(G^n_{X_k}\) and discriminator \(D^n_{X_k}\) at each scale \(n\). For the last convolutional block, we do not use batch normalization and a \(tanh\) is used instead of a LeakyReLU. An adam optimizer learning rate of 0.0005 and parameters \(\beta_1 = 0.5\) and \(\beta_2 = 0.999\), are used. In Eq. 11 and Eq. 18, \(\alpha = 100.0\) for all experiments. \(r\) is chosen to be 0.75 and \(N\) is chosen such that the maximal resolution at the finest scale \(N\) is \(64 \times 64\). The computational cost and time measurements of our method is given in the supplementary (Sec. 4.)

### 4. Experiments

Images are resized to a resolution of \(64 \times 64\). For ten independent trials and for each class label, we train our model on \(k\) different images randomly selected from the normal class of images \(X\). The same set of images are used for our method and for baselines. At test time, for each trial, using all test images, we measure AUC. We report the mean AUC and standard deviation values in the main text, and the detailed results per class in the supplementary (Sec. 2.).

#### 4.1. Anomaly detection

We evaluate our method on four datasets: Paris [26], CIFAR10 [16], FashionMNIST [38] and MNIST [18]. The Paris dataset consists of 6,392 high resolution (1024 × 768) images obtained from Flicker and divided into 11 landmarks. CIFAR10 consists of 60,000 32 × 32 color images in 10 classes, split to 50,000/10,000 between train and test. MNIST and FashionMNIST both consist of 70,000 28×28 grayscale images of digits or fashion products, respectively, split to 60,000/10,000 between train and test. There are 10 categories in each. Following [33, 3, 31], we consider as normal images, all images from a given class, and as anomalous images, the images of all other classes.

We compare our method to five recent baseline methods. The first method is GEOM [10] which applies different transformations to normal images and trains a classifier to classify the transformation applied. The second work is GOAD [3], which modifies the anomaly score used by GEOM. We also consider DROCC [11], that trains a classifier to distinguish the training samples from their perturbations generated adversarially. Lastly, we consider the method of DeepSVDD [31], which uses a similar objective to that of classic SVDD [34] together with features of a deep network, and PatchSVDD [39] which extends DeepSVDD to a patch-based method using self-supervised learning.

**One shot anomaly detection** Fig. 2(a) shows the result of our method in comparison to baselines. Our model outperforms all baselines for all datasets. It scores best for FashionMNIST and MNIST, where a single image captures significant inter-class variability. PatchSVDD, which incorporates a patch-based formulation, scores significantly lower, indicating that, using patches at different scales, together with a generative and discriminative formulation, significantly improves results. DeepSVDD scores better than PatchSVDD, indicating that deep network features are important. DROCC, GEOAM and GOAD, which use a discriminative objective, are also inferior to our method, indicating that our patch based generative model is important.

Qualitative results for the Paris dataset on the one-shot setting are presented in Fig. 3, for typical samples from the “Defense”, “Louvre” and “Pantheon” classes. For each class we train the model on random training image and show: (1) the training image, (2) two randomly generated samples, (3) true positive, true negative, false positive and false negative predictions. As can be seen, the generated images capture the appearance and texture of the class, but sometimes alter the structure. From the true positives, we observe that the model correctly classifies non trivial images that differ from the training image in color, orientation, zoom and even partial occlusions. The false positive (anomalous classified as normal) are often similar to the training sample. The false negatives are often images with significant occlusions or images where the landmark is very small and distant.

**Few shot anomaly detection** We evaluate our method on five and ten images. Fig. 2(b-c) shows the result of our method in comparison to baselines. Our model outperforms the baselines on all datasets, on both the five-shot and ten-shot settings. Increasing the number of samples boosts performance, in particular for the Paris dataset. where using five samples, instead of one, increased the AUC from 66.6% to 79.8%, indicating that multiple images are required to handle inter-class variability. On this dataset, our gap to baselines is also largest, indicating that our method
better utilizes the additional variability provided by the additional samples. In Fig. 4 we consider the effect of increasing the number of samples used for training our method and baselines, on CIFAR10. Our method consistently improves with added number of samples and outperforms baselines on both the 50-shot and 80-shot settings. However, the gap from GEOM decreases as \( k \) increases, indicating that when many samples are available, the improvement gained from generating samples at multiple scales diminishes.

### 4.2. Defect detection

For the task of defect detection, we evaluate our method on the MVTec dataset [4], which contains 5354 high-resolution color images of 10 object and 15 texture categories. These are split to 3629/1725 images between train and test. The number of training samples per category ranges from 60 to 320. A total of 70 defect types, such as little cracks, deformations, discolorizations and scratches occur in test images. The anomalies may differ in size, shape and structure. For each class, we consider the normal class to be all the defect-free images in this class, and the anomalous images to be all the defective images from the same class.

We compare our method to DifferNet [29] and PatchSVDD [39] that excel in defect detection. We also compare to anomaly detection methods described above. We follow the same evaluation protocol for one-shot and few
Figure 3. Illustration of classification decisions made by a single-shot model trained on the Paris dataset. The first column is the training sample, then are random samples generated by the trained generative model. The other columns present samples from the test set of the Paris dataset that are either true positive (**TP**), true negative (**TN**), false positive (**FP**) or false negative (**TN**).

Figure 4. The effect of increasing the number of samples for our method and baselines on CIFAR10.

shot anomaly detection. We consider the standard set of transformations as described in Sec. 3.1 (Ours1 in Fig. 5). For a fair comparison with DifferNet we also consider only the four rotation transformations (group (4) in Sec. 3.1) as applied in DifferNet (Ours2 in Fig. 5). As can be seen in Fig. 5, our method outperforms all baselines. Fig. 6 illustrates the localization of the defects for one-shot defect detection, for our method and for DifferNet. Three random test samples from MVTec are shown. We visualize the defects at the final scale. Following the notation of Sec. 3.3, our visualization map is defined as

$$P_{M_i} = \sum_{i=1}^{M} T^{-1}_{i} [D^{N_i}(T_i(x^N_{test})))_{i}$$

where $D^{N_i}(T_i(x^N_{test}))_{i}$ is an $H \times H$ map indicating how real $T_i(x^N_{test})$’s patches are. We consider only the patch indices in $f^N_{i}(H \times H)$, and set the other patches to 0. For DifferNet, we use the visualization procedure of their work. We accurately capture the defect regions whereas DifferNet only captures smaller regions of the defect area. Additional visualisations are given in the supplementary (Sec. 3.4).

4.3. Ablation analysis

Our method relies on three main components: (1) a generative model, (2) its hierarchical multi-scale nature, and (3) a transformation-discriminating component. We assess the contribution of these components separately, running ablation experiments on CIFAR10 for both one-shot and five-shot settings. A first variant (variant (a), or v.(a) for short) does not have a generative component (G). Instead, we only use $D^{x,n}_{n}$, and remove the fake class $0$. $D^{x,n}_{n}$ is trained to classify between real images at this scale and their transformations (no fake images are used). The anomaly score remains the same. The second variant (v.(b)) does not employ transformations discriminatively (T). $D^{x,n}_{n}$ is trained to distinguish between real and fake images at scale $n$, and not between transformations of images. This is equivalent to setting $M = 1$ and using $T_1$ as identity. As another alternative (v.(c)), we apply $T_1, \ldots, T_M$ as augmentations (uniformly at random) before being fed to $D^{x,n}_{n}$. The next variants consider a single scale of the hierarchy (H), by setting $N = 0$. We use the same $11 \times 11$ receptive field, and downscale the image to either $100 \times 100$, where small patches are considered, or to $20 \times 20$, where large patches are considered. This is indicated by $s = 100$ (v.(d)) and $s = 20$ (v.(e)). We also consider a simple baseline where no component is used. The anomaly score is the MSE between the test image and the training image in the one-shot setting, and the average MSE in the five-shot setting (v.(f)). Finally, variant v.(g), trains a GEOM [10] model on 6,000 samples

<table>
<thead>
<tr>
<th>v.</th>
<th>G</th>
<th>T</th>
<th>H</th>
<th>1-shot</th>
<th>5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>64.9</td>
<td>67.5</td>
</tr>
<tr>
<td>(a)</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>60.7</td>
<td>64.9</td>
</tr>
<tr>
<td>(b)</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>59.1</td>
<td>60.0</td>
</tr>
<tr>
<td>(c)</td>
<td>Yes</td>
<td>Augment</td>
<td>Yes</td>
<td>59.7</td>
<td>63.4</td>
</tr>
<tr>
<td>(d)</td>
<td>Yes</td>
<td>Yes</td>
<td>No $(s=100)$</td>
<td>57.6</td>
<td>57.8</td>
</tr>
<tr>
<td>(e)</td>
<td>Yes</td>
<td>Yes</td>
<td>No $(s=20)$</td>
<td>57.3</td>
<td>63.8</td>
</tr>
<tr>
<td>(f)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>47.7</td>
<td>48.0</td>
</tr>
<tr>
<td>(g)</td>
<td>Generation followed by GEOM</td>
<td>58.8</td>
<td>63.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 5. Average AUC (STD indicated with a vertical line) for defect detection on MVTec, for the One Shot, Five Shot and Ten Shot settings. For Ours1, the transformations of anomaly detection (Sec. 3.1) are used, while for Ours2, only rotations are used (as in DifferNet).

Figure 6. Localization of defects in MVTec test images for one-shot defect detection. (a) The original images, in which the anomaly region is delineated in red. (b-c) The localization provided by (b) our method and (c) DifferNet.

Figure 7. Effect of percentage of patches used for defect detection. Generated using our generative model.

The results are reported in Tab. 4.2. All three components (G, T and H) are required to achieve best performance, with H (hierarchy of patches) playing a particularly important role. Applying GEOM to images generated by our networks is also not as effective as our method. However, it is more effective than running GEOM in the few shot setting (Fig. 2).

We further analyze the effect of the percentage of patches (\(f_i^2(H \times H)\)) taken for defect detection (See Sec. 3.3). Fig. 7 gives the average AUC for MVTec as a function of percentage of patches. Using 5% (resp. 10%) of patches is best for the one-shot (resp. five/ten shot) setting. However, the results are stable, when this ratio remains low. Additional ablation on the anomaly score, the transformations and the generated samples is provided in the supplementary (Sec. 3).

5. Conclusions

We present a multi-scale hierarchical generative model, which incorporates, within the discriminators, the self-supervised task of classifying transformations. While multiclass descriptors, in the supervised case, are common in conditional GANs, e.g., [32], we are not aware of other methods that combine labels from a SSL task. Also, while discriminators play an important role in adversarial learning, most works do not employ them outside of training a generator or for creating a secondary feature matching loss [27]. Also unique, as far as we can ascertain, is the training of single-image like GANs on multiple images. This is done by adding a conditioning layer that contains the image index.

Our method presents a very sizable gap in performance in comparison to the state of the art methods for the few-shot case. Admittedly, training becomes more involved with the increase in the number of training images, and the method does not scale well to hundreds or thousands of training images, without further modifications. In the case of a small training set, for which our method was designed, it demonstrates one-class classification capabilities that are surprising given the emphasis in the existing literature on modeling the form of the variability between the training samples.

6. Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC CoG 725974). The contribution of the first author is part of a Ph.D. thesis research conducted at Tel Aviv University.
References