



Image Anomaly Detection with Generative Adversarial Networks

Lucas Deecke^{1(✉)}, Robert Vandermeulen², Lukas Ruff³, Stephan Mandt⁴,
and Marius Kloft²

¹ University of Edinburgh, Edinburgh, Scotland, UK

`l.deecke@ed.ac.uk`

² TU Kaiserslautern, Kaiserslautern, Germany

`{vandermeulen,kloft}@cs.uni-kl.de`

³ Hasso Plattner Institute, Potsdam, Germany

`lukas.ruff@hpi.de`

⁴ University of California, Irvine, CA, USA

`mandt@uci.edu`

Abstract. Many anomaly detection methods exist that perform well on low-dimensional problems however there is a notable lack of effective methods for high-dimensional spaces, such as images. Inspired by recent successes in deep learning we propose a novel approach to anomaly detection using generative adversarial networks. Given a sample under consideration, our method is based on searching for a good representation of that sample in the latent space of the generator; if such a representation is not found, the sample is deemed anomalous. We achieve state-of-the-art performance on standard image benchmark datasets and visual inspection of the most anomalous samples reveals that our method does indeed return anomalies.

1 Introduction

Given a collection of data it is often desirable to automatically determine which instances of it are unusual. Commonly referred to as anomaly detection, this is a fundamental machine learning task with numerous applications in fields such as astronomy [11, 43], medicine [5, 46, 51], fault detection [18], and intrusion detection [15, 19]. Traditional algorithms often focus on the low-dimensional regime and face difficulties when applied to high-dimensional data such as images or speech. Second to that, they require the manual engineering of features.

Deep learning omits manual feature engineering and has become the de-facto approach for tackling many high-dimensional machine learning tasks. This is largely a testament of its experimental performance: deep learning has helped to achieve impressive results in image classification [24], and is setting new standards in domains such as natural language processing [25, 50] and speech recognition [3].

L. Deecke and R. Vandermeulen—Equal contributions.

© Springer Nature Switzerland AG 2019

M. Berlingerio et al. (Eds.): ECML PKDD 2018, LNAI 11051, pp. 3–17, 2019.

https://doi.org/10.1007/978-3-030-10925-7_1

In this paper we present a novel deep learning based approach to anomaly detection which uses generative adversarial networks (GANs) [17]. GANs have achieved state-of-the-art performance in high-dimensional generative modeling. In a GAN, two neural networks – the discriminator and the generator – are pitted against each other. In the process the generator learns to map random samples from a low-dimensional to a high-dimensional space, mimicking the target dataset. If the generator has successfully learned a good approximation of the training data’s distribution it is reasonable to assume that, for a sample drawn from the data distribution, there exists some point in the GAN’s latent space which, after passing it through the generator network, should closely resembles this sample. We use this correspondence to perform anomaly detection with GANs (ADGAN).

In Sect. 2 we give an overview of previous work on anomaly detection and discuss the modeling assumptions of this paper. Section 3 contains a description of our proposed algorithm. In our experiments, see Sect. 4, we both validate our method against traditional methods and showcase ADGAN’s ability to detect anomalies in high-dimensional data.

2 Background

Here we briefly review previous work on anomaly detection, touch on generative models, and highlight the methodology of GANs.

2.1 Related Work

Anomaly Detection. Research on anomaly detection has a long history with early work going back as far as [12], and is concerned with finding unusual or *anomalous* samples in a corpus of data. An extensive overview over traditional anomaly detection methods as well as open challenges can be found in [6]. For a recent empirical comparison of various existing approaches, see [13].

Generative models yield a whole family of anomaly detectors through estimation of the data distribution p . Given data, we estimate $\hat{p} \approx p$ and declare those samples which are unlikely under \hat{p} to be anomalous. This guideline is roughly followed by traditional non-parametric methods such as kernel density estimation (KDE) [40], which were applied to intrusion detection in [53]. Other research targeted mixtures of Gaussians for active learning of anomalies [42], hidden Markov models for registering network attacks [39], and dynamic Bayesian networks for traffic incident detection [48].

Deep Generative Models. Recently, variational autoencoders (VAEs) [22] have been proposed as a deep generative model. By optimizing over a variational lower bound on the likelihood of the data, the parameters of a neural network are tuned in such a way that samples resembling the data may be generated from a Gaussian prior. Another generative approach is to train a pair of deep convolutional neural networks in an autoencoder setup (DCAE) [33] and producing

samples by decoding random points on the compression manifold. Unfortunately, none of these approaches yield a tractable way of estimating p . Our approach uses a deep generative model in the context of anomaly detection.

Deep Learning for Anomaly Detection. Non-parametric anomaly detection methods suffer from the curse of dimensionality and are thus often inadequate for the interpretation and analysis of high-dimensional data. Deep neural networks have been found to obviate many problems that arise in this context. As a hybrid between the two approaches, deep belief networks were coupled with one-class support vector machines to detect anomalies in [14]. We found that this technique did not work well for image datasets, and indeed the authors included no such experiments in their paper.

A recent work proposed an end-to-end deep learning approach, aimed specifically at the task of anomaly detection [45]. Similarly, one may employ a network that was pretrained on a different task, such as classification on ImageNet [8], and then use this network’s intermediate features to extract relevant information from images. We tested this approach in our experimental section.

GANs, which we discuss in greater depth in the next section, have garnered much attention with its performance surpassing previous deep generative methods. Concurrently to this work, [46] developed an anomaly detection framework that uses GANs in a similar way as we do. We discuss the differences between our work and theirs in Sect. 3.2.

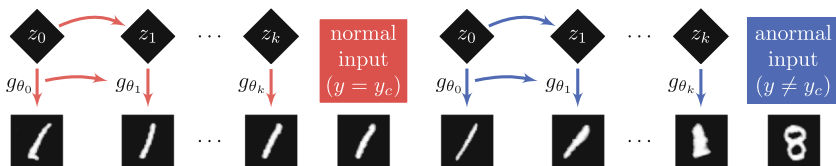


Fig. 1. An illustration of ADGAN. In this example, ones from MNIST are considered normal ($y_c = 1$). After an initial draw from p_z , the loss between the first generation $g_{\theta_0}(z_0)$ and the image x whose anomaly we are assessing is computed. This information is used to generate a consecutive image $g_{\theta_1}(z_1)$ more alike x . After k steps, samples are scored. If x is similar to the training data (red example, $y = y_c$), then a similar object should be contained in the image of g_{θ_k} . For a dissimilar x (blue example, $y \neq y_c$), no similar image is found, resulting in a large loss. (Color figure online)

2.2 Generative Adversarial Networks

GANs, which lie at the heart of ADGAN, have set a new state-of-the-art in generative image modeling. They provide a framework to generate samples that are approximately distributed to p , the distribution of the training data $\{x_i\}_{i=1}^n \triangleq \mathcal{X} \subseteq \mathbb{R}^d$. To achieve this, GANs attempt to learn the parametrization of a neural network, the so-called generator g_θ , that maps low-dimensional

samples drawn from some simple noise prior p_z (e.g. a multivariate Gaussian) to samples in the image space, thereby inducing a distribution q_θ (the push-forward of p_z with respect to g_θ) that approximates p . To achieve this a second neural network, the discriminator d_ω , learns to classify the data from p and q_θ . Through an alternating training procedure the discriminator becomes better at separating samples from p and samples from q_θ , while the generator adjusts θ to fool the discriminator, thereby approximating p more closely. The objective function of the GAN framework is thus:

$$\min_{\theta} \max_{\omega} \left\{ V(\theta, \omega) = \mathbb{E}_{x \sim p} [\log d_\omega(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - d_\omega(g_\theta(z)))] \right\}, \quad (1)$$

where z are vectors that reside in a latent space of dimensionality $d' \ll d$.¹ A recent work showed that this minmax optimization (1) equates to an empirical lower bound of an f -divergence [37].²

GAN training is difficult in practice, which has been shown to be a consequence of vanishing gradients in high-dimensional spaces [1]. These instabilities can be countered by training on integral probability metrics (IPMs) [35, 49], one instance of which is the 1-Wasserstein distance.³ This distance, informally defined, is the amount of work to pull one density onto another, and forms the basis of the Wasserstein GAN (WGAN) [2]. The objective function for WGANs is

$$\min_{\theta} \max_{\omega \in \Omega} \left\{ W(\theta, \omega) = \mathbb{E}_{x \sim p} [d_\omega(x)] - \mathbb{E}_{z \sim p_z} [d_\omega(g_\theta(z))] \right\}, \quad (2)$$

where the parametrization of the discriminator is restricted to allow only 1-Lipschitz functions, i.e. $\Omega = \{\omega : \|d_\omega\|_{\text{L}} \leq 1\}$. When compared to classic GANs, we have observed that WGAN training is much more stable and is thus used in our experiments, see Sect. 4.

3 Algorithm

Our proposed method (ADGAN, see Algorithm 1) sets in after GAN training has converged. If the generator has indeed captured the distribution of the training data then, given a new sample $x \sim p$, there should exist a point z in the latent space, such that $g_\theta(z) \approx x$. Additionally we expect points away from the support of p to have no representation in the latent space, or at least occupy a small portion of the probability mass in the latent distribution, since they are easily discerned by d_ω as not coming from p . Thus, given a test sample x , if there exists no z such that $g_\theta(z) \approx x$, or if such a z is difficult to find, then it can

¹ That p may be approximated via transformations from a low-dimensional space is an assumption that is implicitly motivated from the manifold hypothesis [36].

² This lower bound becomes tight for an optimal discriminator, making apparent that $V(\theta, \omega^*) \propto \text{JS}[p|q_\theta]$.

³ This is achieved by restricting the class over which the IPM is optimized to functions that have Lipschitz constant less than one. Note that in Wasserstein GANs, an expression corresponding to a lower bound is optimized.

be inferred that x is not distributed according to p , i.e. it is anomalous. Our algorithm hinges on this hypothesis, which we illustrate in Fig. 1.

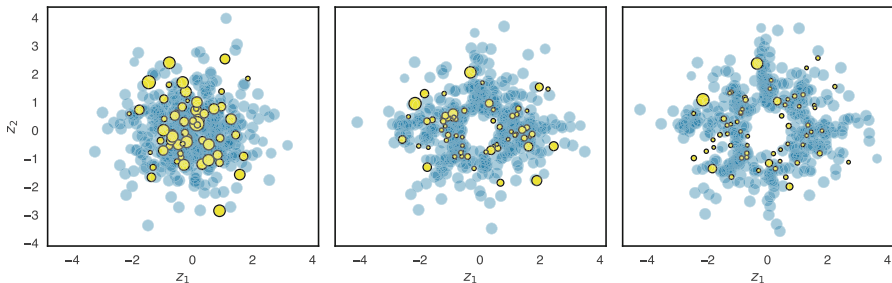


Fig. 2. The coordinates (z_1, z_2) of 500 samples from MNIST are shown, represented in a latent space with $d' = 2$. At different iterations t of ADGAN, no particular structure arises in the z -space: samples belonging to the normal class (\bullet) and the anomalous class (\circ) are scattered around freely. Note that this behavior also prevents $p_z(z_t)$ from providing a sensible anomaly score. The sizes of points correspond to the reconstruction loss between generated samples and their original image $\ell(g_\theta(z_t), x)$. The normal and anomalous class differ markedly in terms of this metric. (Color figure online)

3.1 ADGAN

To find z , we initialize from $z_0 \sim p_z$, where p_z is the same noise prior also used during GAN training. For $t = 1, \dots, k$ steps, we backpropagate the reconstruction loss ℓ between $g_\theta(z_t)$ and x , making the subsequent generation $g_\theta(z_{t+1})$ more like x . At each iteration, we also allow a small amount of flexibility to the parametrization of the generator, resulting in a series of mappings from the latent space $g_{\theta_0}(z_0), \dots, g_{\theta_k}(z_k)$ that more and more closely resembles x . Adjusting θ gives the generator additional representative capacity, which we found to improve the algorithm’s performance. Note that these adjustments to θ are not part of the GAN training procedure and θ is reset back to its original trained value for each new testing point.

To limit the risk of seeding in unsuitable regions and address the non-convex nature of the underlying optimization problem, the search is initialized from n_{seed} individual points. The key idea underlying ADGAN is that if the generator was trained on the same distribution x was drawn from, then the average over the final set of reconstruction losses $\{\ell(x, g_{\theta_{j,k}}(z_{j,k}))\}_{j=1}^{n_{\text{seed}}}$ will assume low values, and high values otherwise. In Fig. 2 we track a collection of samples through their search in a latent space of dimensionality $d' = 2$.

Our method may also be understood from the standpoint of approximate inversion of the generator. In this sense, the above backpropagation finds latent vectors z that lie close to $g_\theta^{-1}(x)$. Inversion of the generator was previously

Algorithm 1. Anomaly Detection using Generative Adversarial Networks (ADGAN).

Input: parameters $(\gamma, \gamma_\theta, n_{\text{seed}}, k)$, sample x , GAN generator g_θ , prior p_z , reconstruction loss ℓ

Initialize $\{z_{j,0}\}_{j=1}^{n_{\text{seed}}} \sim p_z$ and $\{\theta_{j,0}\}_{j=1}^{n_{\text{seed}}} \triangleq \theta$
for $j = 1$ **to** n_{seed} **do**
 for $t = 1$ **to** k **do**
 $z_{j,t} \leftarrow z_{j,t-1} - \gamma \cdot \nabla_{z_{j,t-1}} \ell(g_{\theta_{j,t-1}}(z_{j,t-1}), x)$
 $\theta_{j,t} \leftarrow \theta_{j,t-1} - \gamma_\theta \cdot \nabla_{\theta_{j,t-1}} \ell(g_{\theta_{j,t-1}}(z_{j,t-1}), x)$
 end for
end for

Return: $(1/n_{\text{seed}}) \sum_{j=1}^{n_{\text{seed}}} \ell(g_{\theta_{j,k}}(z_{j,k}), x)$

studied in [7], where it was verified experimentally that this task can be carried out with high fidelity. In addition [29] showed that generated images can be successfully recovered by backpropagating through the latent space.⁴ Jointly optimizing latent vectors and the generator parametrization via backpropagation of reconstruction losses was investigated in detail by [4]. The authors found that it is possible to train the generator entirely without a discriminator, still yielding a model that incorporates many of the desirable properties of GANs, such as smooth interpolations between samples.

3.2 Alternative Approaches

Given that GAN training also gives us a discriminator for discerning between real and fake samples, one might reasonably consider directly applying the discriminator for detecting anomalies. However, once converged, the discriminator exploits checkerboard-like artifacts on the pixel level, induced by the generator architecture [31, 38]. While it perfectly separates real from forged data, it is not equipped to deal with samples which are completely unlike the training data. This line of reasoning is verified in Sect. 4 experimentally.

Another approach we considered was to evaluate the likelihood of the final latent vectors $\{z_{j,k}\}_{j=1}^{n_{\text{seed}}}$ under the noise prior p_z . This approach was tested experimentally in Sect. 4, and while it showed some promise, it was consistently outperformed by ADGAN.

In [46], the authors propose a technique for anomaly detection (called AnGAN) which uses GANs in a way somewhat similar to our proposed algorithm. Their algorithm also begins by training a GAN. Given a test point x , their algorithm searches for a point z in the latent space such that $g_\theta(z) \approx x$ and computes the reconstruction loss. Additionally they use an intermediate

⁴ While it was shown that any $g_\theta(z)$ may be reconstructed from some other $z_0 \in \mathbb{R}^{d'}$, this does not mean that the same holds for an x not in the image of g_θ .

Table 1. ROC-AUC of classic anomaly detection methods. For both MNIST and CIFAR-10, each model was trained on every class, as indicated by y_c , and then used to score against remaining classes. Results for KDE and OC-SVM are reported both in conjunction with PCA, and after transforming images with a pre-trained Alexnet.

DATASET	y_c	KDE		OC-SVM		IF	GMM	DCAE	AnoGAN	VAE	ADGAN
		PCA	ALEXNET	PCA	ALEXNET						
MNIST	0	0.982	0.634	0.993	0.962	0.957	0.970	0.988	0.990	0.884	0.999
	1	0.999	0.922	1.000	0.999	1.000	0.999	0.993	0.998	0.998	0.992
	2	0.888	0.654	0.881	0.925	0.822	0.931	0.917	0.888	0.762	0.968
	3	0.898	0.639	0.931	0.950	0.924	0.951	0.885	0.913	0.789	0.953
	4	0.943	0.676	0.962	0.982	0.922	0.968	0.862	0.944	0.858	0.960
	5	0.930	0.651	0.881	0.923	0.859	0.917	0.858	0.912	0.803	0.955
	6	0.972	0.636	0.982	0.975	0.903	0.994	0.954	0.925	0.913	0.980
	7	0.933	0.628	0.951	0.968	0.938	0.938	0.940	0.964	0.897	0.950
	8	0.924	0.617	0.958	0.926	0.814	0.889	0.823	0.883	0.751	0.959
	9	0.940	0.644	0.970	0.969	0.913	0.962	0.965	0.958	0.848	0.965
		0.941	0.670	0.951	0.958	0.905	0.952	0.919	0.937	0.850	0.968
CIFAR-10	0	0.705	0.559	0.653	0.594	0.630	0.709	0.656	0.610	0.582	0.661
	1	0.493	0.487	0.400	0.540	0.379	0.443	0.435	0.565	0.608	0.435
	2	0.734	0.582	0.617	0.588	0.630	0.697	0.381	0.648	0.485	0.636
	3	0.522	0.531	0.522	0.575	0.408	0.445	0.545	0.528	0.667	0.488
	4	0.691	0.651	0.715	0.753	0.764	0.761	0.288	0.670	0.344	0.794
	5	0.439	0.551	0.517	0.558	0.514	0.505	0.643	0.592	0.493	0.640
	6	0.771	0.613	0.727	0.692	0.666	0.766	0.509	0.625	0.391	0.685
	7	0.458	0.593	0.522	0.547	0.480	0.496	0.690	0.576	0.516	0.559
	8	0.595	0.600	0.719	0.630	0.651	0.646	0.698	0.723	0.522	0.798
	9	0.490	0.529	0.475	0.530	0.459	0.384	0.705	0.582	0.633	0.643
		0.590	0.570	0.587	0.601	0.558	0.585	0.583	0.612	0.524	0.634

discriminator layer d'_ω and compute the loss between $d'_\omega(g_\theta(z))$ and $d'_\omega(x)$. They use a convex combination of these two quantities as their anomaly score.

In ADGAN we never use the discriminator, which is discarded after training. This makes it easy to couple ADGAN with any GAN-based approach, e.g. LSGAN [32], but also any other differentiable generator network such as VAEs or moment matching networks [27]. In addition, we account for the non-convexity of the underlying optimization by seeding from multiple areas in the latent space. Lastly, during inference we update not only the latent vectors z , but jointly update the parametrization θ of the generator.

4 Experiments

Here we present experimental evidence of the efficacy of ADGAN. We compare our algorithm to competing methods on a controlled, classification-type task and show anomalous samples from popular image datasets. Our main findings are that ADGAN:

- outperforms non-parametric as well as available deep learning approaches on two controlled experiments where ground truth information is available;

- may be used on large, unsupervised data (such as LSUN bedrooms) to detect anomalous samples that coincide with what we as humans would deem unusual.

4.1 Datasets

Our experiments are carried out on three benchmark datasets with varying complexity: (i) MNIST [26] which contains grayscale scans of handwritten digits. (ii) CIFAR-10 [23] which contains color images of real world objects belonging to ten classes. (iii) LSUN [52], a dataset of images that show different scenes (such as bedrooms, bridges, or conference rooms). For all datasets the training and test splits remain as their default. All images are rescaled to assume pixel values in $[-1, 1]$.

4.2 Methods and Hyperparameters

We tested the performance of ADGAN against four traditional, non-parametric approaches commonly used for anomaly detection: (i) KDE [40] with a Gaussian kernel. The bandwidth is determined from maximum likelihood estimation over ten-fold cross validation, with $h \in \{2^0, 2^{1/2}, \dots, 2^4\}$. (ii) One-class support vector machine (OC-SVM) [47] with a Gaussian kernel. The inverse length scale is selected with automated tuning, as proposed by [16], and we set $\nu = 0.1$. (iii) Isolation forest (IF) [30], which was largely stable to changes in its parametrization. (iv) Gaussian mixture model (GMM). We allowed the number of components to vary over $\{2, 3, \dots, 20\}$ and selected suitable hyperparameters by evaluating the Bayesian information criterion.

For the methods above we reduced the feature dimensionality before performing anomaly detection. This was done via PCA [41], varying the dimensionality over $\{20, 40, \dots, 100\}$; we simply report the results for which best performance on a small holdout set was attained. As an alternative to a linear projection, we evaluated the performance of both methods after applying a non-linear transformation to the image data instead via an Alexnet [24], pretrained on ImageNet. Just as on images, the anomaly detection is carried out on the representation in the final convolutional layer of Alexnet. This representation is then projected down via PCA, as otherwise the runtime of KDE and OC-SVM becomes problematic.

We also report the performance of two end-to-end deep learning approaches: VAEs and DCAEs. For the DCAE we scored according to reconstruction losses, interpreting a high loss as indicative of a new sample differing from samples seen during training. In VAEs we scored by evaluating the evidence lower bound (ELBO). We found this to perform much better than thresholding directly via the prior likelihood in the latent space or other more exotic approaches, such as scoring from the variance of the inference network.

In both DCAEs and VAEs we use a convolutional architecture similar to that of DCGAN [44], with batch normalization [20] and ReLU activations in each layer. We also report the performance of AnoGAN. To put it on equal footing,

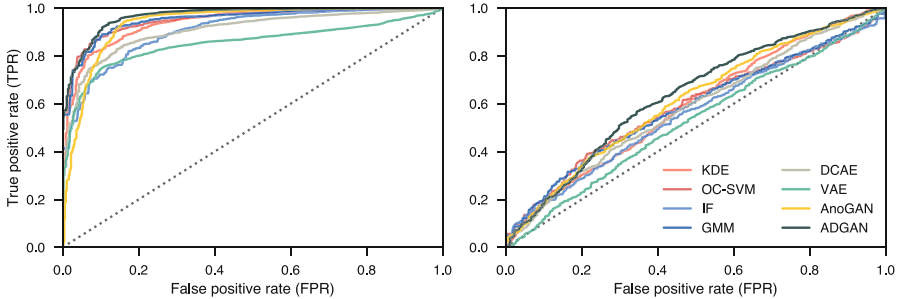


Fig. 3. ROC curves for one-versus-all prediction of competing methods on MNIST (left) and CIFAR10 (right), averaged over all classes. KDE and OC-SVM are shown in conjunction with PCA, for detailed performance statistics see Table 1.

we pair it with DCGAN [44], the same architecture also used for training in our approach.

ADGAN requires a trained generator. For this purpose, we trained on the WGAN objective (2), as this was much more stable than using GANs. The architecture was fixed to that of DCGAN [44]. Following [34] we set the dimensionality of the latent space to $d' = 256$.

For ADGAN, the searches in the latent space were initialized from the same noise prior that the GAN was trained on (in our case a normal distribution). To take into account the non-convexity of the problem, we seeded with $n_{\text{seed}} = 64$ points. For the optimization of latent vectors and the parameters of the generator we used the Adam optimizer [21].⁵ When searching for a point in the latent space to match a test point, we found that more iterations helped the performance, but this gain saturates quickly. As a trade-off between execution time and accuracy we found $k = 5$ to be a good value, and used this in the results we report. Unless otherwise noted, we measured reconstruction quality with a squared L_2 loss.

4.3 One-Versus-All Classification

The first task is designed to quantify the performance of competing methods. The experimental setup closely follows the original publication on OC-SVMs [47] and we begin by training models on data from a single class from MNIST. Then we evaluate each model’s performance on 5000 items randomly selected from the test set, which contains samples from all classes. In each trial, we label the classes unseen in training as anomalous.

Ideally, a method assigns images from anomalous classes (say, digits 1-9) a higher anomaly score than images belonging to the normal class (zeros). Varying the decision threshold yields the receiver operating characteristic (ROC), shown

⁵ From a quick parameter sweep, we set the learning rate to $\gamma = 0.25$ and $(\beta_1, \beta_2) = (0.5, 0.999)$. We update the generator with $\gamma_\theta = 5 \cdot 10^{-5}$, the default learning rate recommended in [2].

in Fig. 3 (left). The second experiment follows this guideline with the colored images from CIFAR-10, and the resulting ROC curves are shown in Fig. 3 (right). In Table 1, we report the AUCs that resulted from leaving out each individual class.

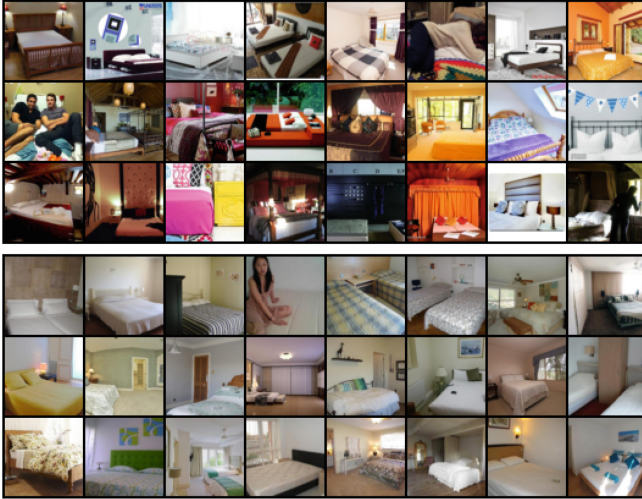


Fig. 4. Starting from the top left, the first three rows show samples contained in the LSUN bedrooms validation set which, according to ADGAN, are the most anomalous (have the highest anomaly score). Again starting from the top left corner, the bottom rows contain images deemed normal (have the lowest score).

In these controlled experiments we highlight the ability of ADGAN to outperform traditional methods at the task of detecting anomalies in a collection of high-dimensional image samples. While neither table explicitly contains results from scoring the samples using the GAN discriminator, we did run these experiments for both datasets. Performance was weak, with an average AUC of 0.625 for MNIST and 0.513 for CIFAR-10. Scoring according to the prior likelihood p_z of the final latent vectors worked only slightly better, resulting in an average AUC of 0.721 for MNIST and 0.554 for CIFAR-10. Figure 2 gives an additional visual intuition as to why scoring via the prior likelihood fails to give a sensible anomaly score: anomalous samples do not get sent to low probability regions of the Gaussian distribution.

4.4 Unsupervised Anomaly Detection

In the second task we showcase the use of ADGAN in a practical setting where no ground truth information is available. For this we first trained a generator on LSUN scenes. We then used ADGAN to find the most anomalous images within the corresponding validation sets containing 300 images. The images associated

with the highest and lowest anomaly scores of three different scene categories are shown in Figs. 4, 5, and 6. Note that the large training set sizes in this experiment would complicate the use of non-parametric methods such as KDE and OC-SVMs.



Fig. 5. Scenes from LSUN showing conference rooms as ranked by ADGAN. The top rows contain anomalous samples, the bottom rows scenes categorized as normal.



Fig. 6. Scenes from LSUN showing churches, ranked by ADGAN. Top rows: anomalous samples. Bottom rows: normal samples.

To additionally quantify the performance on LSUN, we build a test set from combining the 300 validation samples of each scene. After training the generator on bedrooms only we recorded whether ADGAN assigns them low anomaly scores, while assigning high scores to samples showing any of the remaining scenes. This resulted in an AUC of 0.641.

As can be seen from visually inspecting the LSUN scenes flagged as anomalous, our method has the ability to discern usual from unusual samples. We infer that ADGAN is able to incorporate many properties of an image. It does not merely look at colors, but also takes into account whether shown geometries are canonical, or whether an image contains a foreign object (like a caption). Opposed to this, samples that are assigned a low anomaly score are in line with a classes' *Ideal Form*. They show plain colors, are devoid of foreign objects, and were shot from conventional angles. In the case of bedrooms, some of the least anomalous samples are literally just a bed in a room.

5 Conclusion

We showed that searching the latent space of the generator can be leveraged for use in anomaly detection tasks. To that end, our proposed method: (i) delivers state-of-the-art performance on standard image benchmark datasets; (ii) can be used to scan large collections of unlabeled images for anomalous samples.

To the best of our knowledge we also reported the first results of using VAEs for anomaly detection. We remain optimistic that boosting its performance is possible by additional tuning of the underlying neural network architecture or an informed substitution of the latent prior.

Accounting for unsuitable initializations by jointly optimizing latent vectors and generator parameterization are key ingredients to help ADGAN achieve strong experimental performance. Nonetheless, we are confident that approaches such as initializing from an approximate inversion of the generator as in ALI [9, 10], or substituting the reconstruction loss for a more elaborate variant, such as the Laplacian pyramid loss [28], can be used to improve our method further.

Acknowledgments. We kindly thank reviewers for their constructive feedback, which helped to improve this work. LD gratefully acknowledges funding from the School of Informatics, University of Edinburgh. LR acknowledges financial support from the German Federal Ministry of Transport and Digital Infrastructure (BMVI) in the project OSIMAB (FKZ: 19F2017E). MK and RV acknowledge support from the German Research Foundation (DFG) award KL 2698/2-1 and from the Federal Ministry of Science and Education (BMBF) award 031B0187B.

References

1. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. In: International Conference on Learning Representations (2017)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223 (2017)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: International Conference on Learning Representations (2015)
4. Bojanowski, P., Joulin, A., Lopez-Paz, D., Szlam, A.: Optimizing the latent space of generative networks. In: International Conference on Machine Learning (2018)
5. Campbell, C., Bennett, K.P.: A linear programming approach to novelty detection. In: Advances in Neural Information Processing Systems, pp. 395–401 (2001)
6. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. *ACM Comput. Surv. (CSUR)* **41**(3), 15 (2009)
7. Creswell, A., Bharath, A.A.: Inverting the generator of a generative adversarial network. arXiv preprint [arXiv:1611.05644](https://arxiv.org/abs/1611.05644) (2016)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
9. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. In: International Conference on Learning Representations (2017)
10. Dumoulin, V., et al.: Adversarially learned inference. In: International Conference on Learning Representations (2017)
11. Dutta, H., Giannella, C., Borne, K., Kargupta, H.: Distributed top-k outlier detection from astronomy catalogs using the DEMAC system. In: International Conference on Data Mining, pp. 473–478. SIAM (2007)
12. Edgeworth, F.: XLI. on discordant observations. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **23**(143), 364–375 (1887)
13. Emmott, A.F., Das, S., Dietterich, T., Fern, A., Wong, W.K.: Systematic construction of anomaly detection benchmarks from real data. In: ACM SIGKDD Workshop on Outlier Detection and Description, pp. 16–21. ACM (2013)
14. Erfani, S.M., Rajasegarar, S., Karunasekera, S., Leckie, C.: High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognit.* **58**, 121–134 (2016)
15. Eskin, E.: Anomaly detection over noisy data using learned probability distributions. In: International Conference on Machine Learning (2000)
16. Evangelista, P.F., Embrechts, M.J., Szymanski, B.K.: Some properties of the gaussian kernel for one class learning. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 269–278. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74690-4_28
17. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
18. Görnitz, N., Braun, M., Kloft, M.: Hidden Markov anomaly detection. In: International Conference on Machine Learning, pp. 1833–1842 (2015)
19. Hu, W., Liao, Y., Vemuri, V.R.: Robust anomaly detection using support vector machines. In: International Conference on Machine Learning, pp. 282–289 (2003)
20. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)

21. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
22. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013)
23. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
24. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
25. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
26. LeCun, Y.: The MNIST database of handwritten digits (1998). <http://yann.lecun.com/exdb/mnist>
27. Li, Y., Swersky, K., Zemel, R.: Generative moment matching networks. In: International Conference on Machine Learning, pp. 1718–1727 (2015)
28. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: Computer Vision and Pattern Recognition, pp. 246–253. IEEE (2006)
29. Lipton, Z.C., Tripathi, S.: Precise recovery of latent vectors from generative adversarial networks. In: International Conference on Learning Representations, Workshop Track (2017)
30. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: International Conference on Data Mining, pp. 413–422. IEEE (2008)
31. Lopez-Paz, D., Oquab, M.: Revisiting classifier two-sample tests. In: International Conference on Learning Representations (2017)
32. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: International Conference on Computer Vision, pp. 2794–2802. IEEE (2017)
33. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. Artificial Neural Networks and Machine Learning (ICANN), pp. 52–59 (2011)
34. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. In: International Conference on Learning Representations (2017)
35. Müller, A.: Integral probability metrics and their generating classes of functions. Adv. Appl. Probab. **29**(2), 429–443 (1997)
36. Narayanan, H., Mitter, S.: Sample complexity of testing the manifold hypothesis. In: Advances in Neural Information Processing Systems, pp. 1786–1794 (2010)
37. Nowozin, S., Cseke, B., Tomioka, R.: f-GAN: training generative neural samplers using variational divergence minimization. In: Advances in Neural Information Processing Systems, pp. 271–279 (2016)
38. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill **1**(10), e3 (2016)
39. Ourston, D., Matzner, S., Stump, W., Hopkins, B.: Applications of hidden Markov models to detecting multi-stage network attacks. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences. IEEE (2003)
40. Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Stat. **33**(3), 1065–1076 (1962)
41. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philos. Mag. **2**(11), 559–572 (1901)
42. Pelleg, D., Moore, A.W.: Active learning for anomaly and rare-category detection. In: Advances in Neural Information Processing Systems, pp. 1073–1080 (2005)

43. Protopapas, P., Giammarco, J., Faccioli, L., Struble, M., Dave, R., Alcock, C.: Finding outlier light curves in catalogues of periodic variable stars. *Mon. Not. R. Astron. Soc.* **369**(2), 677–696 (2006)
44. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint [arXiv:1511.06434](https://arxiv.org/abs/1511.06434) (2015)
45. Ruff, L., et al.: Deep one-class classification. In: *International Conference on Machine Learning* (2018)
46. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Niethammer, M., et al. (eds.) *IPMI 2017*. LNCS, vol. 10265, pp. 146–157. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59050-9_12
47. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Technical report MSR-TR-99-87, Microsoft Research (1999)
48. Singliar, T., Hauskrecht, M.: Towards a learning traffic incident detection system. In: *Workshop on Machine Learning Algorithms for Surveillance and Event Detection, International Conference on Machine Learning* (2006)
49. Sriperumbudur, B.K., Fukumizu, K., Gretton, A., Schölkopf, B., Lanckriet, G.R.: On integral probability metrics, ϕ -divergences and binary classification. arXiv preprint [arXiv:0901.2698](https://arxiv.org/abs/0901.2698) (2009)
50. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: *Advances in Neural Information Processing Systems*, pp. 3104–3112 (2014)
51. Wong, W.K., Moore, A.W., Cooper, G.F., Wagner, M.M.: Bayesian network anomaly pattern detection for disease outbreaks. In: *International Conference on Machine Learning*, pp. 808–815 (2003)
52. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from abbey to zoo. In: *Computer Vision and Pattern Recognition*, pp. 3485–3492. IEEE (2010)
53. Yeung, D.Y., Chow, C.: Parzen-window network intrusion detectors. In: *International Conference on Pattern Recognition*, vol. 4, pp. 385–388. IEEE (2002)