24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Regularized Cycle Consistent Generative Adversarial Network for Anomaly Detection

Ziyi Yang[1] and **Iman Soltani Bozchalooi**[2] and **Eric Darve**[3]

Abstract. In this paper, we investigate algorithms for anomaly detection. Previous anomaly detection methods focus on modeling the distribution of non-anomalous data provided during training. However, this does not necessarily ensure the correct detection of anomalous data. We propose a new Regularized Cycle Consistent Generative Adversarial Network (RCGAN) in which deep neural networks are adversarially trained to better recognize anomalous samples. This approach is based on leveraging a penalty distribution with a new definition of the loss function and novel use of discriminator networks. It is based on a solid mathematical foundation, and proofs show that our approach has stronger guarantees for detecting anomalous examples compared to the current state-of-the-art. Experimental results on both real-world and synthetic data show that our model leads to significant and consistent improvements on previous anomaly detection benchmarks. Notably, RCGAN improves on the state-of-the-art on the KDDCUP, Arrhythmia, Thyroid, Musk and CIFAR10 datasets.

1 INTRODUCTION

Anomaly detection refers to the task of identifying anomalous observations that deviate from what are believed to be normal data. It has been an important and active research area in many domains, such as medical diagnosis [21], cyber intrusion detection [3] and robotics [14]. Emerging deep learning models [8] with extraordinary capacity to estimate the complex distributions in high-dimensional data provide new approaches for anomaly detection. Efforts have been made to address anomaly detection by deep neural networks, for example energy-based model [24] and deep Gaussian mixture model [25].

Recently proposed bi-directional Generative Adversarial Networks (GANs), including Adversarial Learned Inference (ALI) [6] and ALI with Conditional Entropy (ALICE) [9], allow for high-quality mapping back from data to latent variable space via an encoder network. Adversarially Learned Anomaly Detection (ALAD) in [23], which is built upon ALICE, leverages reconstruction of data from GAN for anomaly detection. Despite inspiring progress in deep anomaly detection, most of the previous work focuses on density estimation based primarily on normal data, i.e. the generation of normal data. However, this does not guarantee the detection of anomalous data. For instance, methods that rely on GAN start making spurious predictions when the generator network is able to (incorrectly) generate points outside the normal manifold. A theoretical understanding on how and why generative models can detect anomaly is still lacking.

[1] Stanford University, USA, email: ziyi.yang@stanford.edu
[2] Ford Greenfield Labs, USA, email: isoltani@ford.com
[3] Stanford University, USA, email: darve@stanford.edu

In this paper, we propose a theoretically grounded algorithm based on GAN with special modifications to the loss function and discriminator networks to bias both the generator and the discriminator towards the normal manifold. We introduce a penalty distribution $t(\boldsymbol{x})$ w.r.t. the normal data distribution $q(\boldsymbol{x})$ in the adversarial training such that the generation from the latent space is biased towards normal manifold and discriminator is trained adversarially to assign higher probability to normal data. The penalty distribution is chosen to be a random noise distribution (e.g., Gaussian or uniform distribution) and the motivation will be explained in section 4. Mathematical proofs show that the introduction of the penalty distribution results in a more consistent detection of anomalous data (avoiding false positive). Results on synthetic data also demonstrate that RCGAN yields more faithful generators and discriminators for anomaly detection than previous GAN-based models.

We evaluate our approach on real-world datasets including KDDCUP (network intrusion), Arrhythmia, Thyroid (medical diagnosis), Musk (molecular chemistry) and CIFAR10 (vision). On all these datasets, RCGAN outperforms all other baseline models by significant margins.

In summary, our key contributions are two-fold:

- The introduction of the penalty distribution $t(\boldsymbol{x})$ to GAN-based framework for anomaly detection to bias the generator and the discriminator towards the normal manifold.
- Mathematical proofs show that RCGAN enforces a large reconstruction error on anomalous data and encourages accurate reconstruction for normal data, providing a theoretical guarantee for reliable anomaly detection.

2 RELATED WORK

Also known as novelty detection and outlier detection, anomaly detection has been extensively studied in literature. Previous methods can be roughly categorized into two types, representation learning and generative model.

Representation learning methods address anomaly detection by extracting common features or learning a data mapping from normal data. One-Class Support Vector Machines (OC-SVM) [22] finds a maximum margin hyperplane such that mapped normal data are separated from the origin. Deep Support Vector Data Description (DSVDD) [18] optimizes a hypershpere to enclose the network representations of the normal data. ODIN [10] utilizes temperature scaling and perturbations upon a pre-trained neural network for image anomaly detection. In [7] researchers develop an approach for vision anomaly detection by training a classifier on geometric-transformed normal images. The classifier essentially provides feature detec-

tors with softmax activation statistics that can be used to compute anomaly scores.

Generative models mostly try to learn the reconstruction of data and detect anomaly through reconstruction profiles. For example, autoencoders are used to model the normal data distribution and the anomaly scores are computed as the reconstruction loss or how likely a sample can be reconstructed [2, 13, 15]. In [19] researchers introduce distorted normal data to the training of autoencoders, however, theoretical analysis is lacking why the method works. Deep Structured Energy Based Models (DSEBM) [24] learn an energy-based model to map each example to an energy score. Deep Autoencoding Gaussian Mixture Model (DAGMM) [25] estimates Gaussian mixture from normal data via an autoencoder network. Recently, Generative Adversarial Networks have been explored for anomaly detection. GANs are leveraged to identify disease markers on tomography images of the retina [21]. Images are mapped back to the latent variable space by a recursive backpropagation process. ALAD [23] adopts a bi-directional GAN framework and data are projected to the latent space by the encoder network.

Among all generative methods, our work is mostly related to ALAD. In contrast to ALAD, RCGAN utilizes samples from an *a priori* chosen random noise distribution as adversarial data during training and enables discriminators to better recognize anomalies, as suggested by our theoretical analysis and experimental evaluations.

3 PRELIMINARIES

In this section we briefly introduce the anomaly detection problem from a statistical angle. Then we go through closely related GAN frameworks and their applications for anomaly detection. Finally, we motivate why the application of penalty distribution in the training of GANs is essential.

3.1 Anomaly Detection from a Statistical Perspective

The anomaly detection problem can be formulated as follows. We consider that the "normal" data is defined by a probability density function $q(\boldsymbol{x})$ ("normal" does not refer to Gaussian distribution in our paper unless mentioned). In unsupervised anomaly detection, during training, we only have access to samples from $q(\boldsymbol{x})$. The goal is to learn an anomaly score function $A(\boldsymbol{x})$ such that, during the **test** phase, anomalous examples are assigned with larger anomaly scores than normal examples.

3.2 Generative Adversarial Networks

One approach for anomaly detection is to model the underlying distribution of normal data $q(\boldsymbol{x})$ based on training normal examples. Generative Adversarial Networks (GANs) [8] can model a distribution using a transformation $G(\boldsymbol{z})$ from a latent space distribution $p(\boldsymbol{z})$ to the space $\boldsymbol{x}$; the *generator* network $G(\boldsymbol{z})$ defines the conditional distribution $p(\boldsymbol{x}|\boldsymbol{z})$. The generator distribution is defined as $p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$.

The GANs framework trains a *discriminator* network $D(\boldsymbol{x})$ to distinguish between real data from $q(\boldsymbol{x})$ and synthetic data generated from $p(\boldsymbol{x})$. The minmax objective function for GANs is:

$$\begin{aligned}\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log D(\boldsymbol{x})] \\ + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]\end{aligned} \tag{1}$$

[8] shows that the optimal generator and discriminator correspond in Eq. (1) to a saddle point such that the generator distribution matches the data distribution $p(\boldsymbol{x}) = q(\boldsymbol{x})$.

Adversarially Learned Inference (ALI) from [6] introduces an *encoder* network $E(\boldsymbol{x})$ and attempts to match the encoder joint distribution $q(\boldsymbol{x}, \boldsymbol{z}) = q(\boldsymbol{x})e(\boldsymbol{z}|\boldsymbol{x})$ and the generator joint distribution $p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$, where $e(\boldsymbol{z}|\boldsymbol{x})$ is parameterized by the encoder network. The same idea is also proposed in [4]. The optimization objective for ALI is defined as:

$$\begin{aligned}\min_{E,G} \max_{D_{\boldsymbol{xz}}} V_{\text{ALI}}(D_{\boldsymbol{xz}}, G, E) = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log D_{\boldsymbol{xz}}(\boldsymbol{x}, E(\boldsymbol{x}))] \\ + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1 - D_{\boldsymbol{xz}}(G(\boldsymbol{z}), \boldsymbol{z}))]\end{aligned} \tag{2}$$

where $D_{\boldsymbol{xz}}$ is a discriminator network that takes both $\boldsymbol{x}$ and $\boldsymbol{z}$ as input and the output is the probability that $\boldsymbol{x}$ and $\boldsymbol{z}$ are from $q(\boldsymbol{x}, \boldsymbol{z})$. It follows that the optimum of the encoder, generator and discriminator is a saddle point of Eq. (2) if and only if $q(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x}, \boldsymbol{z})$. Also if a solution of Eq. (2) is achieved, the marginal and joint distributions in $(\boldsymbol{x}, \boldsymbol{z})$ match.

In order to address the non-identifiability issues in ALI, [9] proposes ALI with Conditional Entropy (ALICE) that adds a second discriminator $D_{\boldsymbol{xx}}$ network to ALI to distinguish $\boldsymbol{x}$ and its reconstruction $\hat{\boldsymbol{x}} = G(E(\boldsymbol{x}))$. An extra term is included in the overall optimization objective, written as:

$$\begin{aligned}\min_{E,G} \max_{D_{\boldsymbol{xz}}, D_{\boldsymbol{xx}}} V_{\text{ALICE}} = V_{\text{ALI}} + \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log D_{\boldsymbol{xx}}(\boldsymbol{x}, \boldsymbol{x}) \\ + \log(1 - D_{\boldsymbol{xx}}(\boldsymbol{x}, \hat{\boldsymbol{x}}))]\end{aligned} \tag{3}$$

[9] shows that Eq. (3) approximates an upper bound of the conditional entropy $H^{\pi}(x|z) = -\mathbb{E}_{\pi(\boldsymbol{x},\boldsymbol{z})}[\log \pi(\boldsymbol{x}|\boldsymbol{z})]$, where $\pi(\boldsymbol{x}, \boldsymbol{z})$ represents the matched joint distribution $\pi(\boldsymbol{x}, \boldsymbol{z}) \triangleq q(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x}, \boldsymbol{z})$. It follows that the corresponding optimal generator and encoder in Eq. (3) theoretically guarantees a perfect reconstruction for $\boldsymbol{x} \sim q(\boldsymbol{x})$.

Recently, efforts have been made to utilize GANs for anomaly detection, especially bi-directional GANs mentioned above that can readily reconstruct a data example via the encoder and generator network. For example, Adversarially Learned Anomaly Detection (ALAD) in [23] trains ALICE model on normal data, where they add an extra discriminator $D_{\boldsymbol{zz}}$ to encourage cycle consistency in the latent space. The anomaly score $A(\boldsymbol{x})$ depends on how well $\boldsymbol{x}$ can be reconstructed.

Although previous generative models have the ability to reconstruct normal samples and assign low anomaly score to normal data, these models offer limited guarantees for detecting anomalous samples. To be more specific, these methods rely on the property that the reconstruction $G(E(\boldsymbol{x}))$ is close to $\boldsymbol{x}$ for normal data and far for anomalous; however, $G(E(\boldsymbol{x}))$ does not necessarily yield poor reconstruction (leading to high anomaly scores) for anomalous samples. For example, as shown in Fig. 1, an autoencoder trained on normal data can wrongly produce highly accurate reconstructions of abnormal data. Also for previous GAN-based models, it is not guaranteed that the discriminator, trained to distinguish between $\boldsymbol{x} \sim q(\boldsymbol{x})$ and "fake" samples from the generator, can successfully discriminate between normal and abnormal data. One example is shown in the third row of Fig. 3 where the discriminators fail to recognize the manifold of normal data.

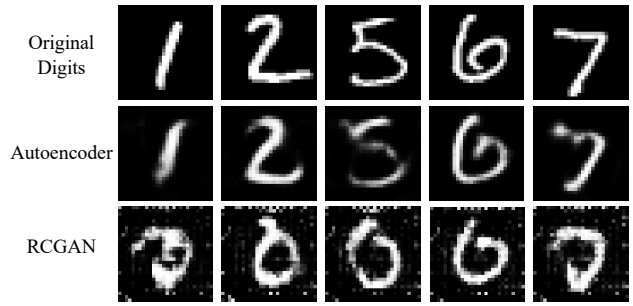

Figure 1. Examples of reconstructed abnormal data by RCGAN and an autoencoder (AE), which both are trained on the normal images of digit 0. The first row includes five examples of original abnormal digits. The second and third row contain reconstructed images by AE and RCGAN respectively. AE (wrongly) reconstructs abnormal data with relatively high accuracy, and this may lead to false positive anomaly detection. In contrast, RCGAN successfully regularizes the generation towards the normal manifold by using $t(\boldsymbol{x})$, with the reconstructed abnormal data resembling digit 0. This result empirically demonstrates that RCGAN offers stronger guarantees for anomaly detection, which is consistent with the theory.

4 METHODOLOGY

4.1 Regularize the Discriminator and Generator

To tackle the limitations mentioned above and enable the GAN to distinguish between normal and abnormal data, we propose a penalty distribution $t(\boldsymbol{x})$ such that $\boldsymbol{x} \sim t(\boldsymbol{x})$ are considered as adversarial examples for the discriminator during training.

Our method uses ALICE as starting point. Concretely, we propose to include a regularization term $\mathbb{E}_{\boldsymbol{x}\sim t(\boldsymbol{x})} \log(1 - D_{\boldsymbol{xz}}(\boldsymbol{x}, E(\boldsymbol{x})))$ in Eq. (2). The first part of our objective function is as follows:

$$\begin{aligned}\min_{E,G} \max_{D_{\boldsymbol{xz}}} V_{\text{ano}}(D_{\boldsymbol{xz}}, G, E) = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log D_{\boldsymbol{xz}}(\boldsymbol{x}, E(\boldsymbol{x}))] \\ + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1 - D_{\boldsymbol{xz}}(G(\boldsymbol{z}), \boldsymbol{z}))] \\ + \mathbb{E}_{\boldsymbol{x}\sim t(\boldsymbol{x})}[\log(1 - D_{\boldsymbol{xz}}(\boldsymbol{x}, E(\boldsymbol{x})))]\end{aligned} \tag{4}$$

In our model, the penalty distribution $t(\boldsymbol{x})$ is chosen to be a random distribution, e.g., Gaussian distribution. These random distributions have broad support so that by adversarial training, the generator and the discriminators are biased towards to the normal manifold as is proved later in this section. As a result, if we consider $G(E(\boldsymbol{x}))$ where $\boldsymbol{x}$ is anomalous, $G(E(\boldsymbol{x}))$ must be close to the normal sample distribution. Consequently, $G(E(\boldsymbol{x}))$ must be far from $\boldsymbol{x}$, which is the desired outcome for our detection algorithm. This is empirically validated by the examples shown in Fig. 1. An RCGAN trained on images of digit 0 successfully biases the generation towards the normal data manifold: the reconstructions of abnormal data resemble digit 0. This leads to large differences between anomalous data and their reconstructions, and consequently a strong guarantee of anomaly detection. These arguments are supported by the theoretical analysis in this section and the superior experimental performance for our model.

Next, we present theoretical results showing that the optimal discriminator and generator distribution are biased more strongly towards normal data. We first derive the optimal discriminator and then the corresponding optimal generator. Consider the following joint distributions:

- The encoder joint distribution on normal data $q(\boldsymbol{x}, \boldsymbol{z}) = q(\boldsymbol{x})e(\boldsymbol{z}|\boldsymbol{x})$.
- The encoder joint distribution on penalty data $t(\boldsymbol{x}, \boldsymbol{z}) = t(\boldsymbol{x})e(\boldsymbol{z}|\boldsymbol{x})$
- The generator joint distribution $p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$

The conditional distributions $p(\boldsymbol{x}|\boldsymbol{z})$ and $e(\boldsymbol{z}|\boldsymbol{x})$ are specified by the generator and the encoder networks respectively. Recall that marginal distributions $q(\boldsymbol{x})$, $t(\boldsymbol{x})$ and $p(\boldsymbol{z})$ correspond to normal data distribution, penalty distribution and latent variable distribution. The following proposition shows the optimal discriminator $D_{\boldsymbol{xz}}$:

Proposition 1 *For fixed generator G and encoder E, the optimal discriminator $D^*_{\boldsymbol{xz}}$ from Eq. (4) is given by:*

$$\begin{aligned}D^*_{\boldsymbol{xz}} &= \frac{q(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{x}, \boldsymbol{z}) + t(\boldsymbol{x}, \boldsymbol{z}) + p(\boldsymbol{x}, \boldsymbol{z})} \\ &= \frac{q(\boldsymbol{x}, \boldsymbol{z})}{(1 + \frac{t(\boldsymbol{x})}{q(\boldsymbol{x})})q(\boldsymbol{x}, \boldsymbol{z}) + p(\boldsymbol{x}, \boldsymbol{z})}\end{aligned} \tag{5}$$

The proof is in the supplementary material (SM). This optimal discriminator considers both normal data distribution and penalty data distribution. This result shows that, unlike classic GANs trained only on normal data, the optimal discriminator in our model is assigning higher probability to more normal data and lower probability to anomalous data with smaller $q(\boldsymbol{x})$. Experiments on synthetic datasets in section 5.1 further support this conclusion.

Next we will show the optimal generator distribution. Substitute Eq. (5) back to Eq. (4) and let $s(\boldsymbol{x}, \boldsymbol{z}) = q(\boldsymbol{x}, \boldsymbol{z}) + t(\boldsymbol{x}, \boldsymbol{z}) + p(\boldsymbol{x}, \boldsymbol{z})$ and $C(E, G) = V(D^*_{\boldsymbol{xz}}, G, E)$ for shorthand, it follows that:

$$\begin{aligned}C(E, G) = 2\, D_{\text{KL}}\Big(\frac{1}{2}(t(\boldsymbol{x}, \boldsymbol{z}) + p(\boldsymbol{x}, \boldsymbol{z})) \parallel \frac{1}{3}s(\boldsymbol{x}, \boldsymbol{z})\Big) \\ + D_{\text{KL}}\Big(q(\boldsymbol{x}, \boldsymbol{z}) \parallel \frac{1}{3}s(\boldsymbol{x}, \boldsymbol{z})\Big) - \log\frac{27}{4}\end{aligned} \tag{6}$$

where D_{KL} denotes Kullback–Leibler divergence.

Theorem 1 *Given any encoder, the optimal generator distribution $p(\boldsymbol{x}, \boldsymbol{z})$ minimizing Eq. (6) is achieved at*

$$p(\boldsymbol{x}_i, \boldsymbol{z}_j) = \max(0, \beta q(\boldsymbol{x}_i, \boldsymbol{z}_j) - t(\boldsymbol{x}_i, \boldsymbol{z}_j)) \tag{7}$$

where

$$\beta = \frac{1 + \sum_{(m,n)\in S_\beta} t(\boldsymbol{x}_m, \boldsymbol{z}_n)}{\sum_{(m,n)\in S_\beta} q(\boldsymbol{x}_m, \boldsymbol{z}_n)} \tag{8}$$

with $S_\beta = \{(m, n) \mid \beta q(\boldsymbol{x}_m, \boldsymbol{z}_n) - t(\boldsymbol{x}_m, \boldsymbol{z}_n) \geq 0\}$. Eq. (8) has a unique solution (note that β shows up on both sides of Eq. (8)), and $1 \leq \beta \leq 2$. Moreover, $\beta = 1$ whenever $qt = 0$ everywhere (i.e., q and n do not overlap), and $\beta = 2$ whenever $2q - t \geq 0$ everywhere (e.g., $q = n$).

The proof follows Karush-Kuhn-Tucke (KKT) conditions and the convex property of β. The detailed explanation is given in the SM.

The optimal generator in Eq. (7) guarantees that anomalous data $\boldsymbol{x}$ with low $q(\boldsymbol{x})$ has a poor reconstruction. Since $\beta q(\boldsymbol{x}_i, \boldsymbol{z}_j) - t(\boldsymbol{x}_i, \boldsymbol{z}_j) = (\beta q(\boldsymbol{x}_i) - t(\boldsymbol{x}_i))e(\boldsymbol{z}_j|\boldsymbol{x}_i)$, this theorem indicates that the optimal generator maps the latent variable $\boldsymbol{z}$ to $\boldsymbol{x}$ for which the normal data probability $q(\boldsymbol{x})$ is high and the penalty distribution $t(\boldsymbol{x})$ is low. The penalty distribution $t(\boldsymbol{x})$ is used to bias the generator more strongly towards the normal set, removing the "outliers" with low $q(\boldsymbol{x})$, as shown in Thm. 1 and Prop. 1, and therefore improving accuracy overall. In the absence of any information about actual anomalies, this is an effective strategy.

As an example, assume that $q(\boldsymbol{x})$ has support in a manifold of dimension less than d in a unit cube (assuming as is commonly the case that the input data is normalized). As a result $q(\boldsymbol{x})$ is large inside this manifold and small outside. In contrast, $t(\boldsymbol{x})$ is chosen to be roughly uniform inside the unit sphere (e.g., a Gaussian distribution with standard deviation 1 and mean 0). In this set up, whenever $q(\boldsymbol{x})$ is large $t(\boldsymbol{x})$ is small (it doesn't have to be close to zero) and vice versa. In that case our algorithm will provide the correct bias for G and will improve the predictions. A simple example in $\mathbb{R}^2$ is shown in Fig. 2. The normal $q(\boldsymbol{x})$ is a "narrow" Gaussian distribution and $t(\boldsymbol{x})$ is roughly uniform in $[-1, 1]^2$. Note the $t(\boldsymbol{x})$ is unrelated to the actual anomalous examples. The resulting generator $p(\boldsymbol{x})$ is computed by minimizing Eq. (6) using the convex optimization library CVXPY [1] (for simplicity of presentation, we assume the latent distribution $p(\boldsymbol{z})$ and the conditional distribution $e(\boldsymbol{z}|\boldsymbol{x})$ to be constant, so $p(\boldsymbol{x}) \sim p(\boldsymbol{x}, \boldsymbol{z})$ and $q(\boldsymbol{x}) \sim q(\boldsymbol{x}, \boldsymbol{z})$). Same as proved in Thm. 1, RCGAN correctly biases the generator distribution towards the normal manifold.

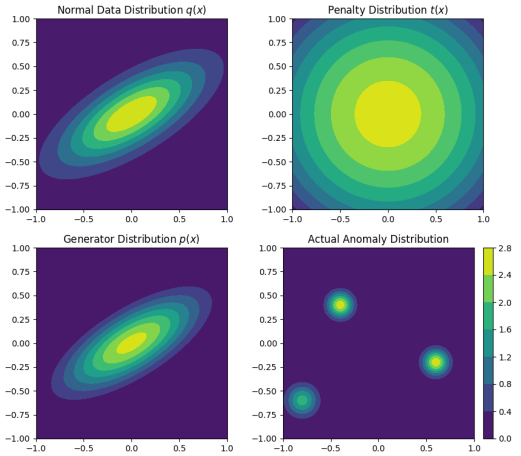

Figure 2. An example of normal data distribution $q(\boldsymbol{x})$, penalty distribution $t(\boldsymbol{x})$, resulting generator distribution in Eq. (7) and actual anomaly distribution. It shows that our algorithm essentially "chops off" regions where $q(\boldsymbol{x})$ is small. This regularization happens independently of how $t(\boldsymbol{x})$ is related to the actual anomaly distribution.

4.2 Generation with Cycle Consistency

To further guarantee a good reconstruction for normal data $\boldsymbol{x} \sim q(\boldsymbol{x})$, we include a second discriminator $D_{\boldsymbol{xx}}$ as in [9] to enforce cycle-consistency of $\boldsymbol{x}$ and its reconstruction. The cycle-consistency optimization objective function is defined as:

$$\min_{E,G} \max_{D_{\boldsymbol{xx}}} V_{\text{cycle}}(D_{\boldsymbol{xx}}, G, E) = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log D_{\boldsymbol{xx}}(\boldsymbol{x}, \boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log(1 - D_{\boldsymbol{xx}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}))] \tag{9}$$

where $\tilde{\boldsymbol{x}} = G(E(\boldsymbol{x}))$ is the reconstruction of $\boldsymbol{x}$. As shown in [9], the optimal generator and encoder of the objective in Eq. (9) leads to $\mathbb{E}_{e(\boldsymbol{z}|\boldsymbol{x})} p(\tilde{\boldsymbol{x}}|\boldsymbol{z}) = \delta(\boldsymbol{x} - \tilde{\boldsymbol{x}})$, resulting in a perfect reconstruction for $\boldsymbol{x} \sim q(\boldsymbol{x})$ theoretically. The optimal discriminator is $D^*_{\boldsymbol{xx}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \delta(\boldsymbol{x} - \tilde{\boldsymbol{x}})$.

The complete minmax optimization objective of the our framework *Regularized Cycle Consistent GAN* (RCGAN) is the sum of Eqns. (4) and (9):

$$\min_{E,G} \max_{D_{\boldsymbol{xz}}, D_{\boldsymbol{xx}}} V_{\text{ano}}(D_{\boldsymbol{xz}}, G, E) + V_{\text{cycle}}(D_{\boldsymbol{xx}}, G, E) \tag{10}$$

After the model is trained on normal data from $p(\boldsymbol{x})$ and adversarial data from $t(\boldsymbol{x})$ following Eq. (10), at the detection phase, the anomaly score assigned to an example $\boldsymbol{x}$ is defined as:

$$A(\boldsymbol{x}) = 1 - D_{\boldsymbol{xx}}(\boldsymbol{x}, G(E(\boldsymbol{x}))) \tag{11}$$

The anomaly score $A(\boldsymbol{x})$ describes how well the example $\boldsymbol{x}$ is reconstructed, determined by the discriminator $D_{\boldsymbol{xx}}$. As showed previously, our model enforces a large reconstruction error on anomalous data with low $q(\boldsymbol{x})$ (which is a desirable feature for identification). Meanwhile, the cycle-consistent objective function in Eq. (9) encourages accurate reconstruction for normal data. This discrepancy endows our model with the ability to discriminate the abnormal from the normal much more reliably. In the next section, numerical experiments on both synthetic and real-world datasets will further demonstrate the effectiveness of our model.

5 EXPERIMENTS

5.1 Towards Better Discriminators

We first test on synthetic dataset in $\mathbb{R}^2 = \{\boldsymbol{x}_1, \boldsymbol{x}_2\}$. We compare our model with another GAN-based method ALAD. Since both RCGAN and ALAD use discriminators for anomaly detection, we plot the discriminators output after training on normal data. Three cases of normal data distributions $q(\boldsymbol{x})$ are tested on: loop, arc and four-dot. The normal data samples are shown in the first row in Fig. 3. The second and the third row contain output by discriminators in RCGAN and ALAD respectively. The output probability given by $D_{\boldsymbol{xz}}(\boldsymbol{x}, E(\boldsymbol{x}))$ and $D_{\boldsymbol{xx}}(\boldsymbol{x}, G(E(\boldsymbol{x})))$ over $\boldsymbol{x}_1 \in [-3, 3]$, $\boldsymbol{x}_2 \in [-3, 3]$ are presented in the first and second column in each case. Higher probability (represented by brighter color) indicates that the discriminator is more confident that $\boldsymbol{x}$ is normal. RCGAN and ALAD are trained using the same neural structures and hyper-parameters for fair comparison. We use the multivariate Gaussian $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ as $t(\boldsymbol{x})$ in RCGAN.

Discriminators from RCGAN exhibit superior performance, assigning lower normal probability to anomalous data outside the normal data cluster. Notice in the "four-dot" case that normal data reside in discontinuous clusters; nevertheless, $D_{\boldsymbol{xz}}$ and $D_{\boldsymbol{xx}}$ in RCGAN accurately recognize anomalous data between normal data clusters. This shows that random noise from the penalty distribution $t(\boldsymbol{x})$, serving as adversarial examples during the training, encourages discriminators to assign low probability to regions where normal examples are missing. This matches with our previous theoretical analysis.

5.2 Baseline Models

In our experiments with real-world dataset, we compare RCGAN with following baseline models:

Anomaly Detection GAN (ADGAN) is the first GAN-based anomaly detection model [21]. After training a DCGAN [16] on normal data, test examples are mapped back to corresponding latent variables $\boldsymbol{z}$ by minimizing the weighted sum of reconstruction error and feature mapping error via gradient descent.

Adversarially Learned Anomaly Detection (ALAD) in [23] uses the GAN framework proposed in ALICE and exploit the encoder network to map data back to latent variable space. The anomaly score is the feature mapping error estimated from $D_{\boldsymbol{xx}}$.

Deep Autoencoding Gaussian Mixture Model (DAGMM) in [25] learns an autoencoder for feature extraction and a Gaussian Mixture Model for density estimation. Data with small weighted sum of

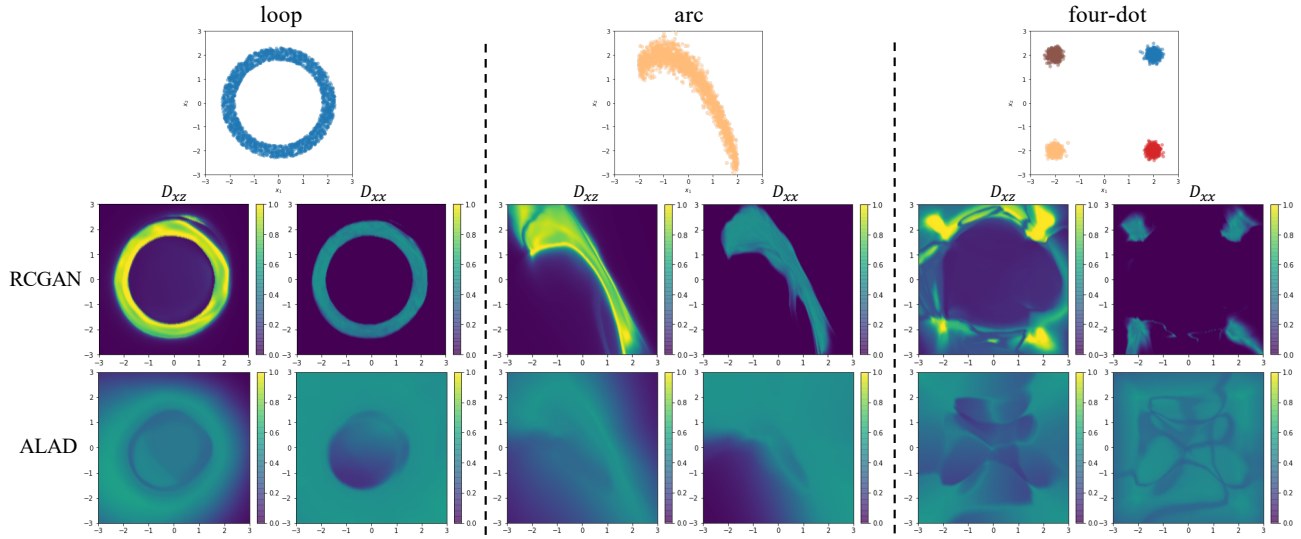

Figure 3. Results on three synthetic datasets: "loop," "arc" and "four-dot." The first row shows samples of normal data, and the second and third row show the output probability of discriminators in RCGAN and ALAD respectively. In each dataset, the left column visualizes the output probability of $D_{\boldsymbol{xz}}$, and the right one shows the output from $D_{\boldsymbol{xx}}$. These plots show the clear distinction between normal and abnormal sets with RCGAN. ALAD's prediction is much fuzzier.

probabilities predicted by the learned Gaussian mixture are considered anomalous.

Deep Structured Energy-Based Model (DSEBM) trains a network that outputs the energy associated with a data example [24]. Two types of DSEBM, leveraging energy (DSEBM-e) and reconstruction error (DSEBM-r) respectively for anomaly detection, are included for comparison.

Deep Support Vector Data Description (DSVDD) trains a neural network while minimizing the volume of a hypersphere that encloses the network representations of the data [18]. The anomaly score is defined as the Euclidean distance of the data the center of hypersphere.

Isolation Forests (IF) constructs trees by randomly selecting features and then arbitrarily choosing a split value on selected features [11]. The anomaly score of an example is defined as the averaged path length from the root node to the example.

One Class Support Vector Machines (OC-SVM) is a kernel-based method that finds maximum margin hyperplane that separates normal data from the origin [22]. We use an RBF kernel $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\frac{1}{m}\|\boldsymbol{x} - \boldsymbol{x}'\|^2)$ in the experiment, where m is the size of input features.

Deep Convolutional Autoencoder (DCAE) is a classical autoencoder with encoder and decoder based on convolutional neural network [12]. The anomaly score is l_2 norm of the reconstruction error.

5.3 Tabular Dataset

We tested on four tabular datasets: KDDCUP, Arrhythmia, Thyroid and MUSK. Note we choose Thyroid and MUSK dataset for their low anomaly ratio (2.5% and 3.2%) to examine RCGAN's robustness in demanding scenario. The experiment setups follow [23, 25]:

- **KDDCUP**. The original KDDCUP network intrusion dataset [5] contains 494,021 samples with 34 categorical and 7 continuous features. During pre-processing, categorical features are encoded using one-hot representation, and the final data examples have 121 dimensions. Data labelled as "non-intrusion" (consisting of 20% in the dataset) are treated as anomalies since they are in a minority group. In the test phase, the top 20% of test data with the highest anomaly scores $A(\boldsymbol{x})$ are predicted as anomalies.
- **Arrhythmia**. The Cardiac Arrhythmia dataset [5] has 452 instances with 274 attributes, and each instance is classified into one of 16 groups. The smallest classes, including 3, 4, 5, 7, 8, 9, 14, and 15, consist of 15% of the entire samples and are treated as the anomaly class. The remaining groups are considered as normal data. The top 15% of the test data with the highest anomaly scores $A(\boldsymbol{x})$ are labeled as anomalies.
- **Thyroid**. The Thyroid disease dataset from [5] is a three-class classification dataset with 3,772 instances and 6 continuous attributes. The "hyperfunction" class, consisting of 2.5% of the dataset set, is treated as anomaly. Therefore, the top 2.5% of the test data with the highest anomaly scores $A(\boldsymbol{x})$ are inferred as abnormal.
- **Musk**. The Musk Anomaly Detection dataset processed by [17] is originally a multi-class classification dataset on musk molecular with 3,062 instances and 166 attributes. The musk category 213 and 211 are regarded as anomalous, and the overall anomaly ratio is 3.2%.

We take 80% of data by random sampling for training and the remaining for test in KDDCUP and Arrhythmia. For Thyroid, 50% of data are randomly chosen for training, and the rest are for test. For Musk, we follow the original data split. Models are evaluated by precision, recall and F1 scores of the anomaly examples predicted. The results are summarized in Table 1. We collect performance of benchmark models from [25] and [23], except that ALAD on Arrhythmia and DSVDD are run by us. Results for RCGAN are averaged over 10 runs. The neural structure for discriminators, generators and encoders used in our models are standard fully connected layers with non-linear gate. We use $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ as $t(\boldsymbol{x})$ in RCGAN.

For a clearer comparison, we also provide the error bar for RCGAN's performance in the last row in table 1. On all four datasets, RCGAN outperforms previous anomaly detection models by significant margins. Especially compared to previous GAN-based methods, the improvement achieved by RCGAN demonstrates the effectiveness of applying a penalty distribution to the adversarial training. On

Table 1. Precision, recall and F_1 in percent on KDDCUP, Arrhythmia, Thyroid and Musk dataset of RCGAN and benchmark models. The last row is the error bar of RCGAN's performance. The best results for each metric are in bold.

Model	KDDCUP			Arrhythmia			Thyroid			Musk		
	Prec.	Recall	F_1	Prec.	Recall	F_1	Prec.	Recall	F_1	Prec.	Recall	F_1
IF	92.16	93.73	92.94	51.47	54.69	53.03	70.13	71.43	70.27	47.96	47.72	47.51
OC-SVM	74.57	85.23	79.54	**53.97**	40.82	45.18	36.39	42.39	38.87	-	-	-
DSEBM-r	85.12	64.72	73.28	15.15	15.13	15.10	4.04	4.03	4.03	-	-	-
DSEBM-e	86.19	64.46	73.99	46.67	45.65	46.01	13.19	13.19	13.19	-	-	-
ADGAN	87.86	82.97	88.65	41.18	43.75	42.42	44.12	46.87	45.45	3.06	3.10	3.10
DAGMM	92.97	94.22	93.69	49.09	50.78	49.83	47.66	48.34	47.82	-	-	-
ALAD	94.27	95.77	95.01	50.00	53.13	51.52	22.92	21.57	22.22	58.16	59.03	58.37
DSVDD	89.81	94.97	92.13	35.32	34.35	34.79	22.22	23.61	23.29	-	-	-
RCGAN	**95.17**	**96.69**	**95.92**	52.73	**56.06**	**54.14**	**77.08**	**75.56**	**76.26**	**67.00**	**66.21**	**66.49**
error bar	0.28	0.29	0.28	6.6	6.8	5.8	4.3	2.7	2.8	5.06	2.53	2.62

Table 2. Novelty detection on CIFAR-10 dataset by treating each class as normal evaluated by AUROC. Performance with highest mean is in bold.

Normal	DCAE	DSEBM	DAGMM	IF	ADGAN	ALAD	**RCGAN**
airplane	59.1±5.1	41.4±2.3	56.0±6.9	60.1±0.7	67.1±2.5	64.7±2.6	**71.8±1.5**
auto.	57.4±2.9	57.1±2.0	56.0±6.9	50.8±0.6	54.7±3.4	45.7±0.8	**59.5±0.7**
bird	48.9±2.4	61.9±0.1	53.8±4.0	49.2±0.4	52.9±3.0	**67.0±0.7**	66.2±0.2
cat	58.4±1.2	50.1±0.4	51.2±0.8	55.1±0.4	54.5±1.9	59.2±1.1	**63.9±1.7**
deer	54.0±1.3	73.2±0.2	52.2±7.3	49.8±0.4	65.1±3.2	72.7±0.6	**73.4±0.9**
dog	**62.2±1.8**	60.5±0.3	49.3±3.6	58.5±0.4	60.3±2.6	52.8±1.2	59.6±1.1
frog	51.2±5.2	68.4±0.3	64.9±1.7	42.9±0.6	58.5±1.4	69.5±1.1	**73.0±1.3**
horse	58.6±2.9	53.3±0.7	55.3±0.8	55.1±0.7	**62.5±0.8**	44.8±0.4	52.5±0.5
ship	**76.8±1.4**	73.9±0.3	51.9±2.4	74.2±0.6	75.8±4.1	73.4±0.4	73.4±3.2
truck	**67.3±3.0**	63.6±3.1	54.2±5.8	58.9±0.7	66.5±2.8	43.2±1.3	57.2±0.6
mean	59.4	60.3	54.4	55.5	61.8	59.3	**65.1**

larger datasets, e.g. KDDCUP, Arrhythmia and MUSK, RCGAN's improvements are statistically significant.

5.4 Image Dataset

We further test on the image dataset CIFAR-10 as a novelty detection task. Ten distinct datasets are generated by regarding each image category as the normal class. We follow the train/test split in the original dataset. The metrics for evaluation is area under the receiver operating curve (AUROC), averaged on 10 runs. We introduce another type of anomaly score function, proposed in ALAD [23]:

$$A(\boldsymbol{x}) = \|l_{\boldsymbol{xx}}(\boldsymbol{x}, \boldsymbol{x}) - l_{\boldsymbol{xx}}(\boldsymbol{x}, G(E(\boldsymbol{x})))\|_2 \tag{12}$$

where $l_{\boldsymbol{xx}}$ denotes the last layer before the logit output in the discriminator $D_{\boldsymbol{xx}}$. $A(\boldsymbol{x})$ is motivated by the matching loss used to stabilize training of GANs in [20]. Again, our model shows an overall strong performance and achieves highest mean performance in five datasets with statistically significant margins. Notably, RCGAN outperforms baseline models on the average performance across 10 datasets (shown in the last row). Compared with previous GAN-based anomaly detection algorithm, ADGAN and ALAD, RCGAN shows overall competitive results. The hyper-parameters and neural structures of RCGAN closely follow ALAD for a fair comparison. More details on training will be provided in the final version.

6 DISCUSSION

Ablation Study. To further demonstrate the effectiveness of leveraging the penalty distribution $t(\boldsymbol{x})$, we test RCGAN with and without $t(\boldsymbol{x})$ in the adversarial training (by removing the last term in Eq. (4)). Results on Thyroid and Arrhythmia dataset are shown in Table 3. Utilizing the penalty distribution shows consistent improvement over the model without $t(\boldsymbol{x})$. This improvement again confirms the effectiveness of our model.

Table 3. Performance of RCGAN with and without leveraging $t(\boldsymbol{x})$ in the adversarial training.

Model	Arrhythmia			Thyroid		
	Prec.	Recall	F_1	Prec.	Recall	F_1
w/o $t(\boldsymbol{x})$	50.00	51.93	50.85	64.06	66.35	64.94
with $t(\boldsymbol{x})$	52.73	56.06	54.14	77.08	75.56	75.80

Table 4. Using different random distributions $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $\mathcal{N}(\mathbf{0}, 2\boldsymbol{I})$ and $\mathcal{U}(-\mathbf{1}, \mathbf{1})$ as $t(\boldsymbol{x})$ for unsupervised learning tasks on KDDCUP and Thyroid dataset.

$t(\boldsymbol{x})$	KDDCUP			Thyroid		
	Prec.	Recall	F_1	Prec.	Recall	F_1
$\mathcal{N}(\mathbf{0}, \boldsymbol{I})$	95.17	96.69	95.92	77.08	75.56	75.80
$\mathcal{N}(\mathbf{0}, 2\boldsymbol{I})$	95.26	96.78	96.01	75.83	75. 93	76.48
$\mathcal{U}(-\mathbf{1}, \mathbf{1})$	93.85	95.45	94.60	75.34	77.76	76.48

Choices of the Penalty Distribution. In all experiments mentioned previously, the penalty distributions $t(\boldsymbol{x})$ are chosen to be Gaussian distribution with zeros mean and identity variance. In this subsection, we evaluate the effect of using different types of random distributions for $t(\boldsymbol{x})$, including two Gaussian distributions $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ (used for experiments in this paper), $\mathcal{N}(\mathbf{0}, 2\boldsymbol{I})$, and a uniform distri-

bution $\mathcal{U}(-\mathbf{1}, \mathbf{1})$. Our model is robust and produces consistent performance improvements using anyone of these random distributions, as summarized in Table 4.

Function of the Penalty Distribution. The penalty distribution introduced in our model is not designed to ensemble actual anomalous data. Our method still works if $t(\boldsymbol{x})$ is not close to real anomalies. We still have good reconstruction for data with high $q(\boldsymbol{x})$, i.e., high certainty that $\boldsymbol{x}$ is normal, and poor reconstruction for data with low $q(\boldsymbol{x})$ (which is favorable). The conclusion from Thm. 1 is unrelated to actual anomalies. In the case that $t(\boldsymbol{x})$ coincides with normal data $q(\boldsymbol{x})$, we can systematically modify $t(\boldsymbol{x})$ appropriately to avoid this issue, since we have samples (training data) from $q(\boldsymbol{x})$.

Potential Limitation. A failure case (weakness) of RCGAN corresponds to situations where the following conditions are met: (1) an overwhelming amount of training data is available, and (2) the real anomalies are very far from $q(\boldsymbol{x})$. With a large amount of training data, we should expect the algorithm to recognize that samples drawn from the tail of $q(\boldsymbol{x})$, but not anomalous, are normal. However, the introduction of $t(\boldsymbol{x})$ "chops off" the tail of $q(\boldsymbol{x})$, leading to potentially wrong predictions. RCGAN was designed for cases when the training data is more limited or noisy and some amount of regularization (provided by $t(\boldsymbol{x})$) is required to improve the prediction.

7 CONCLUSION

In this paper, we propose a GAN-based anomaly detection approach which explicitly introduces the penalty distribution in the adversarial learning. Theoretical analysis shows that the introduction of the penalty distribution offers stronger guarantees that our model will correctly distinguish normal from abnormal data. In our numerical experiments, we show that our model consistently outperforms baseline anomaly detection models on four tabular datasets and ten image datasets. We also demonstrated that the performance of the algorithm is relatively insensitive to the choice of the penalty distribution. For future work, we would like to extend the usage of penalty distribution to other generative models besides GAN-based framework.

ACKNOWLEDGEMENTS

This research is funded by Ford Motor Company. We also would like to thank anonymous reviewers for their valuable feedback.

A Proofs

A.1 Proof of Proposition 1

In Eq. (4), the discriminator D_{xz} is trained to maximize the quantity $V_{\text{ano}}(D_{\boldsymbol{xz}}, G, E)$, which can be rewritten using the three joint distributions above:

$$\min_{E,G}\max_{D_{\boldsymbol{xz}}} V_{\text{ano}}(D_{\boldsymbol{xz}}, G, E) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}\sim q(\boldsymbol{x},\boldsymbol{z})} \log D_{\boldsymbol{xz}}(\boldsymbol{x},\boldsymbol{z}) +$$
$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{z}\sim p(\boldsymbol{x},\boldsymbol{z})} \log(1 - D_{\boldsymbol{xz}}(\boldsymbol{x},\boldsymbol{z})) + \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}\sim t(\boldsymbol{x},\boldsymbol{z})} \log(1 - D_{\boldsymbol{xz}}(\boldsymbol{x},\boldsymbol{z})))$$
$$= \int_{\boldsymbol{x},\boldsymbol{z}} q(\boldsymbol{x},\boldsymbol{z}) \log D_{\boldsymbol{xz}} + \int_{\boldsymbol{x},\boldsymbol{z}} (t(\boldsymbol{x},\boldsymbol{z}) + p(\boldsymbol{x},\boldsymbol{z})) \log(1 - D_{\boldsymbol{xz}}) \tag{13}$$

Recall that the function $a\log(x)+b\log(1-x)$ achieves its maximum at $x = \frac{a}{a+b}$. Therefore, for fixed generator G and encoder E, the optimal discriminator $D_{\boldsymbol{x},\boldsymbol{z}}$ is:

$$D^*_{\boldsymbol{x},\boldsymbol{z}} = \frac{q(\boldsymbol{x},\boldsymbol{z})}{(1 + \frac{t(\boldsymbol{x})}{q(\boldsymbol{x})})q(\boldsymbol{x},\boldsymbol{z}) + p(\boldsymbol{x},\boldsymbol{z})} \tag{14}$$

A.2 Proof of Theorem 1

Substitute Eq. (5) back to Eq. (4) and let $s(\boldsymbol{x},\boldsymbol{z}) = q(\boldsymbol{x},\boldsymbol{z}) + t(\boldsymbol{x},\boldsymbol{z}) + p(\boldsymbol{x},\boldsymbol{z})$ and $C(E,G) = V(D^*_{\boldsymbol{xz}}, G, E)$ for shorthand, we have:

$$\begin{aligned} C(E,G) &= \int_{\boldsymbol{x},\boldsymbol{z}} q(\boldsymbol{x},\boldsymbol{z}) \log \frac{q(\boldsymbol{x},\boldsymbol{z})}{\frac{1}{3}s(\boldsymbol{x},\boldsymbol{z})} + 2\log 2 - 3\log 3 \\ &\quad + 2\int_{\boldsymbol{x},\boldsymbol{z}} \frac{t(\boldsymbol{x},\boldsymbol{z}) + p(\boldsymbol{x},\boldsymbol{z})}{2} \log \frac{\frac{1}{2}(t(\boldsymbol{x},\boldsymbol{z}) + p(\boldsymbol{x},\boldsymbol{z}))}{\frac{1}{3}s(\boldsymbol{x},\boldsymbol{z})} \\ &= 2\, D_{\text{KL}}(\frac{t(\boldsymbol{x},\boldsymbol{z}) + p(\boldsymbol{x},\boldsymbol{z})}{2} \| \frac{s(\boldsymbol{x},\boldsymbol{z})}{3}) \\ &\quad + D_{\text{KL}}(q(\boldsymbol{x},\boldsymbol{z}) \| \frac{s(\boldsymbol{x},\boldsymbol{z})}{3}) + 2\log 2 - 3\log 3 \end{aligned} \tag{15}$$

where $s(\boldsymbol{x},\boldsymbol{z}) = t(\boldsymbol{x},\boldsymbol{z}) + p(\boldsymbol{x},\boldsymbol{z}) + q(\boldsymbol{x},\boldsymbol{z})$. Consider the discrete case, and let p_{ij}, q_{ij} and t_{ij} denote $p_{\boldsymbol{x}_i,\boldsymbol{z}_j}$, $q_{\boldsymbol{x}_i,\boldsymbol{z}_j}$ and $t_{\boldsymbol{x}_i,\boldsymbol{z}_j}$. Given fixed encoder network, the optimization problem in Eq. (15) can be rewritten as:

$$\begin{aligned} &\min_{p_{ij}} \sum_{i,j}(t_{ij} + p_{ij})\log(t_{ij} + p_{ij}) - \sum_{i,j} q_{ij}\log(t_{ij} + p_{ij} + q_{ij}) \\ &- \sum_{i,j}(t_{ij} + p_{ij})\log(t_{ij} + p_{ij} + q_{ij}) \\ &\text{s.t.} \quad p_{ij} \geq 0, \ \sum_{i,j} p_{ij} = 1 \end{aligned} \tag{16}$$

The Lagrangian is:

$$\begin{aligned} \mathcal{L} &= \lambda\,(\sum_{i,j} p_{ij} - 1) - \sum_{i,j}\mu_{ij}p_{ij} + \sum_{i,j}(t_{ij} + p_{ij})\log(t_{ij} + p_{ij}) \\ &- \sum_{i,j}(t_{ij} + p_{ij})\log(t_{ij} + p_{ij} + q_{ij}) - \sum_{i,j} q_{ij}\log(t_{ij} + p_{ij} + q_{ij}) \end{aligned} \tag{17}$$

where λ and $\{\mu_{ij}\}$ are Karush–Kuhn–Tucker (KKT) multipliers. Following KKT conditions, we have:

$$\frac{\partial \mathcal{L}}{\partial p_{ij}} = 0, \quad \mu_{ij}p_{ij} = 0, \quad \mu_{ij} \geq 0 \tag{18}$$

For all i, j, the first condition in Eq. (18) leads to:

$$\log \frac{t_{ij} + p_{ij}}{t_{ij} + p_{ij} + q_{ij}} + \lambda - \mu_{ij} = 0 \tag{19}$$

If $p_{ij} \neq 0$, then $\mu_{ij} = 0$ and it becomes:

$$p_{ij} = \beta q_{ij} - t_{ij} \tag{20}$$

where $\beta = \frac{1}{e^{\lambda}-1}$. Recall $\sum_{i,j} p_{ij} = 1$, then $\beta = \frac{1+\sum t_{mn}}{\sum q_{mn}}$ for $p_{mn} \neq 0$.

Next, we will derive the condition for $p_{ij} = 0$. If $p_{ij} = 0$, from Eq. (19) we have:

$$\mu_{ij} = \log\frac{t_{ij}}{t_{ij} + q_{ij}} + \lambda \geq 0, \quad q_{ij} \leq (e^{\lambda} - 1)t_{ij}, \quad q_{ij} \leq \frac{1}{\beta}t_{ij} \tag{21}$$

The inequality in the first line is from the KKT conditions. In sum, in the discrete case, the optimal generator distribution given any encoder is:

$$p(\boldsymbol{x}_i,\boldsymbol{z}_j) = \max(0, \beta q(\boldsymbol{x}_i,\boldsymbol{z}_j) - t(\boldsymbol{x}_i,\boldsymbol{z}_j)) \tag{22}$$

where

$$\beta = \frac{1 + \sum_{(m,n)\in S_\beta} t(\boldsymbol{x}_m, \boldsymbol{z}_n)}{\sum_{(m,n)\in S_\beta} q(\boldsymbol{x}_m, \boldsymbol{z}_n)} \tag{23}$$

with $S_\beta = \{(m,n) \mid \beta q(\boldsymbol{x}_m, \boldsymbol{z}_n) - t(\boldsymbol{x}_m, \boldsymbol{z}_n) \geq 0\}$. We now prove a few additional properties of the solution. The set S_1 must be non-empty. For $\beta \geq 1$, the set S_β increases monotonously and therefore cannot be empty. Denote

$$f(\beta) = \frac{1 + \sum_{(m,n)\in S_\beta} t(\boldsymbol{x}_m, \boldsymbol{z}_n)}{\sum_{(m,n)\in S_\beta} q(\boldsymbol{x}_m, \boldsymbol{z}_n)}$$

Since

$$\sum_{(m,n)\in S_\beta} t(\boldsymbol{x}_m, \boldsymbol{z}_n) \geq 0, \qquad \sum_{(m,n)\in S_\beta} q(\boldsymbol{x}_m, \boldsymbol{z}_n) \leq 1 \tag{24}$$

So $f(\beta) \geq 1$. All solutions $\beta = f(\beta)$ therefore satisfy $\beta \geq 1$.

Denote $S^{\beta,\mathrm{c}}$ the complement of the set S^β. Then:

$$f(\beta) = \frac{1 + \sum_{(m,n)\in S^\beta} t(\boldsymbol{x}_m, \boldsymbol{z}_n)}{\sum_{(m,n)\in S^\beta} q(\boldsymbol{x}_m, \boldsymbol{z}_n)} = \frac{2 - \sum_{(m,n)\in S^{\beta,\mathrm{c}}} t(\boldsymbol{x}_m, \boldsymbol{z}_n)}{\sum_{(m,n)\in S^\beta} q(\boldsymbol{x}_m, \boldsymbol{z}_n)}$$
$$\leq \frac{2 - \beta \sum_{(m,n)\in S^{\beta,\mathrm{c}}} q(\boldsymbol{x}_m, \boldsymbol{z}_n)}{\sum_{(m,n)\in S^\beta} q(\boldsymbol{x}_m, \boldsymbol{z}_n)} = \beta + \frac{2 - \beta}{\sum_{(m,n)\in S^\beta} q(\boldsymbol{x}_m, \boldsymbol{z}_n)}$$

Consequently, $f(2) \leq 2$, and all solutions $\beta = f(\beta)$ are such that $\beta \leq 2$.

The function

$$S_p(\beta) = \sum_{mn} \max(0, \beta q(\boldsymbol{x}_m, \boldsymbol{z}_n) - t(\boldsymbol{x}_m, \boldsymbol{z}_n))$$

is a continuous convex and monotonically increasing function of β (although not differentiable). But we have:

$$S_p(\beta) = 1 + (\beta - f(\beta)) \sum_{(m,n)\in S^\beta} q(\boldsymbol{x}_m, \boldsymbol{z}_n)$$

Since $S_p(1) \leq 1$, $S_p(2) \geq 1$, and S_p continuous convex and strictly increasing, then there exists a unique β ($1 \leq \beta \leq 2$) for which $S_p(\beta) = 1$. Therefore, the equation $\beta = f(\beta)$ has a unique solution, and $1 \leq \beta \leq 2$.

If the solution $\beta = 2$, then from the definition of f we see that $S^{\beta,\mathrm{c}}$ must be empty. So $2q - t \geq 0$ everywhere. The converse is true. If $\beta = 1$, then $q \neq 0$ implies $t = 0$ (so that $f(1)$ can be equal to 1; see Eq. (24)). For all indices, either q or t is 0 at the corresponding index. This can be written as $qt = 0$ everywhere, or equivalently the support of q and t do not overlap.

REFERENCES

[1] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd, 'A rewriting system for convex optimization problems', *Journal of Control and Decision*, **5**(1), 42–60, (2018).

[2] Jinwon An and Sungzoon Cho, 'Variational autoencoder based anomaly detection using reconstruction probability', *Special Lecture on IE*, **2**, 1–18, (2015).

[3] Anna L Buczak and Erhan Guven, 'A survey of data mining and machine learning methods for cyber security intrusion detection', *IEEE Communications Surveys & Tutorials*, **18**(2), 1153–1176, (2016).

[4] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell, 'Adversarial feature learning', *arXiv preprint arXiv:1605.09782*, (2016).

[5] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[6] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville, 'Adversarially learned inference', *arXiv preprint arXiv:1606.00704*, (2016).

[7] Izhak Golan and Ran El-Yaniv, 'Deep anomaly detection using geometric transformations', in *Advances in Neural Information Processing Systems*, pp. 9781–9791, (2018).

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, 'Generative adversarial nets', in *Advances in neural information processing systems*, pp. 2672–2680, (2014).

[9] Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin, 'Alice: Towards understanding adversarial learning for joint distribution matching', in *Advances in Neural Information Processing Systems*, pp. 5495–5503, (2017).

[10] Shiyu Liang, Yixuan Li, and R Srikant, 'Enhancing the reliability of out-of-distribution image detection in neural networks', *arXiv preprint arXiv:1706.02690*, (2017).

[11] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou, 'Isolation forest', in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. IEEE, (2008).

[12] Alireza Makhzani and Brendan J Frey, 'Winner-take-all autoencoders', in *Advances in neural information processing systems*, pp. 2791–2799, (2015).

[13] Duc Tam Nguyen, Zhongyu Lou, Michael Klar, and Thomas Brox, 'Anomaly detection with multiple-hypotheses predictions', *arXiv preprint arXiv:1810.13292v4*, (2019).

[14] Daehyung Park, Zackory Erickson, Tapomayukh Bhattacharjee, and Charles C Kemp, 'Multimodal execution monitoring for anomaly detection during robot manipulation', in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 407–414. IEEE, (2016).

[15] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto, 'Generative probabilistic novelty detection with adversarial autoencoders', in *Advances in Neural Information Processing Systems*, pp. 6822–6833, (2018).

[16] Alec Radford, Luke Metz, and Soumith Chintala, 'Unsupervised representation learning with deep convolutional generative adversarial networks', *arXiv preprint arXiv:1511.06434*, (2015).

[17] Shebuti Rayana. Odds library, 2016.

[18] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft, 'Deep one-class classification', in *International Conference on Machine Learning*, pp. 4390–4399, (2018).

[19] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli, 'Adversarially learned one-class classifier for novelty detection', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3379–3388, (2018).

[20] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, 'Improved techniques for training gans', in *Advances in neural information processing systems*, pp. 2234–2242, (2016).

[21] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs, 'Unsupervised anomaly detection with generative adversarial networks to guide marker discovery', in *International Conference on Information Processing in Medical Imaging*, pp. 146–157. Springer, (2017).

[22] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt, 'Support vector method for novelty detection', in *Advances in neural information processing systems*, pp. 582–588, (2000).

[23] Houssam Zenati, Manon Romain, Chuan Sheng Foo, Bruno Lecouat, and Vijay R. Chandrasekhar, 'Adversarially learned anomaly detection', *2018 IEEE International Conference on Data Mining (ICDM)*, 727–736, (2018).

[24] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang, 'Deep structured energy based models for anomaly detection', *arXiv preprint arXiv:1605.07717*, (2016).

[25] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen, 'Deep autoencoding gaussian mixture model for unsupervised anomaly detection', in *International Conference on Learning Representations*, (2018).