P-KDGAN: Progressive Knowledge Distillation with GANs for One-class Novelty Detection

Zhiwei Zhang[1,2*], **Shifeng Chen**[1†] and **Lei Sun**[2]

[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[2]School of Information and Electronics, Beijing Institute of Technology, China
{zw.zhang3, shifeng.chen}@siat.ac.cn, sunlei@bit.edu.cn

Abstract

One-class novelty detection is to identify anomalous instances that do not conform to the expected normal instances. In this paper, the Generative Adversarial Networks (GANs) based on encoder-decoder-encoder pipeline are used for detection and achieve state-of-the-art performance. However, deep neural networks are too over-parameterized to deploy on resource-limited devices. Therefore, Progressive Knowledge Distillation with GANs (P-KDGAN) is proposed to learn compact and fast novelty detection networks. The P-KDGAN is a novel attempt to connect two standard GANs by the designed distillation loss for transferring knowledge from the teacher to the student. The progressive learning of knowledge distillation is a two-step approach that continuously improves the performance of the student GAN and achieves better performance than single step methods. In the first step, the student GAN learns the basic knowledge totally from the teacher via guiding of the pre-trained teacher GAN with fixed weights. In the second step, joint fine-training is adopted for the knowledgeable teacher and student GANs to further improve the performance and stability. The experimental results on CIFAR-10, MNIST, and FMNIST show that our method improves the performance of the student GAN by 2.44%, 1.77%, and 1.73% when compressing the computation at ratios of 24.45:1, 311.11:1, and 700:1, respectively.

1 Introduction

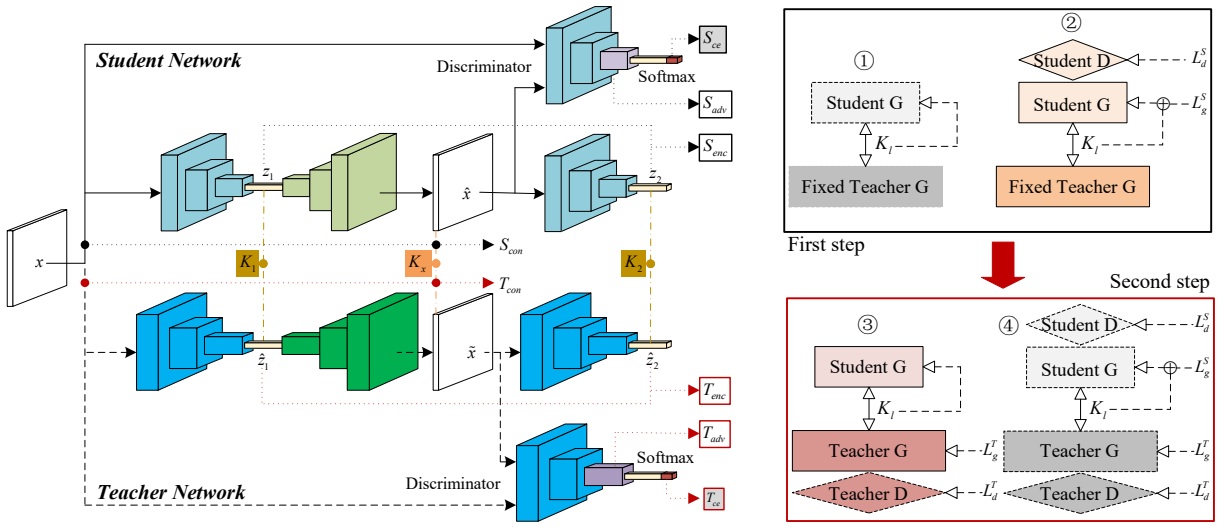
One-class novelty detection aims to identify patterns that do not belong to the normal data distribution [Chandola *et al.*, 2009]. Unlike traditional classification problem, novelty detection is usually trained in an unsupervised setting where novelty data is absent. Novelty detection has a wide variety of applications such as network intrusion [García-Teodoro *et al.*, 2009], credit card fraud [Srivastava *et al.*, 2008], medical diagnoses [Schlegl *et al.*, 2017] and many more. With the advantage of deep learning, novelty detection based on generative adversarial networks (GANs) has shown state-of-the-art performance by learning the representative latent space of high-dimensional data [Schlegl *et al.*, 2017; Zenati *et al.*, 2018; Perera *et al.*, 2019]. However, deep neural networks with high computational costs and large storage prohibit their deployment to computation and memory resource limited systems.

For tackling the above issue, neural network compression has been widely applied in recent years [Cheng *et al.*, 2017]. As one of the mainstream compression methods, Knowledge Distillation (KD) following a teacher-student paradigm transfers knowledge from a teacher network with higher performance to a student network. The early contributions used the outputs of the softmax layers or intermediate layers in teacher networks to improve the performance of student networks [Hinton *et al.*, 2015; Romero *et al.*, 2015]. In the later researches, the discriminator losses were proposed to evaluate the distinction between the distribution spaces of teacher and student networks [Wang *et al.*, 2018a; Wang *et al.*, 2018b; Liu *et al.*, 2018]. To our knowledge, there is no related works on two standard GANs [Goodfellow *et al.*, 2014] including two generators and two discriminators to design distillation loss for knowledge distillation. Additionally, there are rare works investigating the initialization of student networks and always random initialization is used. Our experiments demonstrate that student networks without "knowledge" reserve (with random initialization) do not mimic the outputs of teacher networks well.

In this paper, we apply GANs in the encoder-decoder-encoder structure [Akçay *et al.*, 2018] for one-class novelty detection, which outperforms the state-of-the-art approaches. In order to deploy the deep neural networks in computation resources limited mobile devices, we propose the Progressive Knowledge Distillation with GANs (P-KDGAN) method to train the lightweight student network. The P-KDGAN approach improves the performance of student GAN by solving the following three problems. 1) How to design a distillation loss to measure the similarity of intermediate representations learned from the teacher GAN and the student GAN? As is shown in the student GAN of Figure 1(a), the generator based on encoder-decoder-encoder pipeline can generate two latent vectors z_1, z_2 and a reconstructed image $\hat{x}$. The

*This work was done when Zhiwei Zhang was a research intern at Multimedia Laboratory of SIAT.

†Corresponding author.



(a) Knowledge Distillation with GANs

(b) The Progressive Learning of KDGAN

Figure 1: The flowchart of Knowledge Distillation with GANs for One-class Novelty Detection. (a) The Knowledge Distillation with Generative Adversarial Networks (KDGAN), in which the distillation losses K_l ($K_l = w_1K_1 + w_xK_x + w_2K_2$) is designed for training the student GAN. (b) The two-step progressive learning of KDGAN is used to continuously improve the performance of the student GAN. KDGAN-①, KDGAN-②, KDGAN-③ and KDGAN-④ are four different distillation structures.

generator is trained by minimizing the weighted sum of S_{con}, S_{enc} and S_{adv}, which have defined in Eq.2-4 [Akçay *et al.*, 2018]. Therefore, the distillation loss K_l described in Eq.6 is designed as the weighted sum of the losses K_1, K_x and K_2, where K_1, K_2 represent the difference between the two latent vectors (z_1 and $\hat{z}_1$, z_2 and $\hat{z}_2$) in the teacher GAN and student GAN, and K_x is the difference of two reconstructed images ($\hat{x}$, $\tilde{x}$). 2) How to combine the distillation loss K_l with existing generator losses L_g^S, L_g^T and discriminator losses L_d^S, L_d^T from the student and teacher GANs to improve the performance of the student GAN? As is illustrated in Figure 1(b), we design four distillation structures (KDGAN-①, KDGAN-②, KDGAN-③ and KDGAN-④) based on different combinations of the above five losses. The difference between them consists of two aspects: on the one hand, whether the weights of the teacher GAN are fixed; on the other hand, whether the distillation loss K_l is combined with the losses L_g^S, L_d^S of the student GAN for knowledge transfer. 3) Whether the designed four distillation structures can make the performance of the student GAN like that of the teacher GAN? If not, how to fix it? Our experimental results demonstrate that the performance of student GANs trained from scratch (or with random initialization) by the above four distillation structures is incomparable with the teacher GAN. Just as learning is a gradual cumulative process, a two-step progressive learning of KDGAN is proposed to continuously improve the performance of the student GAN. In the first step, the student GAN imitates the representation of the pre-trained teacher GAN with fixed weights to make itself have a certain "knowledge" reserve. Such a "teaching by teacher" step make the student learn the basic knowledge totally from the teacher. In the second step, the student GAN with basic "knowledge" reserve is fine-trained together with the teacher GAN. The second step of "fine-learning with teacher" can further improve the performance and stability by jointly utilizing the basic knowledge of the teacher and student.

The performance of proposed progressive knowledge distillation with GANs for one-class novelty detection is evaluated on CIFAR-10 [Krizhevsky, 2009], MNIST [LeCun and Cortes, 2005] and FMNIST [Xiao *et al.*, 2017] datasets. Our contributions are summarized as follows.

- We utilize the encoder-decoder-encoder based GAN for one-class novelty detection, which outperforms all the state-of-the-art methods.
- We propose new distillation losses on latent vectors and reconstructed images of GANs that allow the student to better learn from the teacher.
- We regard the distillation process as a knowledgeable teacher to improve the performance and stability of student networks through two-step progressive learning, which includes basic knowledge learning and fine-learning.
- Progressive Knowledge Distillation with GANs is proposed for one-class novelty detection. Our experiments demonstrate that the P-KDAGN can improve the performance of the student GAN on the three datasets CIFAR-10, MNIST and FMNIST by 2.44%, 1.77%, and 1.73%, respectively.

2 Related Work

We briefly review the related works in term of one-class novelty detection and knowledge distillation, as well as the architecture of Ganomaly [Akçay *et al.*, 2018].

2.1 One-class Novelty Detection

In the unsupervised one-class novelty detection, only the normal samples with one class are used for training the model. Conventionally, novelty detection methods can be divided into two categories [Chandola *et al.*, 2009]. One is the traditional methods, such as One-Class SVM (OC-SVM) [Schölkopf *et al.*, 2001], Kernel Density Estimation (KDE) [Parzen, 1962] and Principal Component Analysis (PCA) [Wold *et al.*, 1987]. The disadvantage of such approaches is that they are not suitable for high-dimensional image data. The other methods based on deep learning include Deep Belief Networks (DBN) [Erfani *et al.*, 2016], Autoencoders (AE) [Vincent *et al.*, 2008] and generative adversarial networks (GANs) [Schlegl *et al.*, 2017; Zenati *et al.*, 2018; Perera *et al.*, 2019].

GANs have shown state-of-the-art performance in modeling complex high-dimensional image distributions [Goodfellow *et al.*, 2014]. Therefore, a lot of GANs based methods have been used for novelty detection [Schlegl *et al.*, 2017; Zenati *et al.*, 2018; Perera *et al.*, 2019]. The reconstruction errors of images or latent vectors are utilized as novelty score, which means that the learned model only reconstructs normal samples well, and shows very low tolerance for novel samples. Schlegl et al. [Schlegl *et al.*, 2017] proposed the first GANs based work, AnoGAN, for novelty detection. In training, the combination of the residual loss on images and discrimination loss on feature maps is minimized to iteratively search the best latent vector. The Efficient GAN [Zenati *et al.*, 2018] based on BiGAN [Donahue *et al.*, 2017] network was proposed for jointly training the map from the image to the latent space simultaneously. Perera et al. [Perera *et al.*, 2019] proposed the OCGAN in which two discriminators were used in the latent space and the input space for making the learned network better model the input images. Recently, Ganomaly [Akçay *et al.*, 2018] shown in Figure 1(a) constructs a novel architecture for multi-class anomaly detection. In our method, the Ganomaly framework is used for one-class novelty detection.

2.2 Knowledge Distillation

To reduce the large computation and storage cost of deep convolutional neural networks, knowledge distillation can transfer the generalization ability of a large network (or an ensemble of networks) to a light-weight network. Hinton et al. [Hinton *et al.*, 2015] used the outputs of the softmax layer of a teacher network as the target function to train the student network. Romero et al. [Romero *et al.*, 2015] proposed that a student network with random initialization can imitate the intermediate representations of the teacher network to improve its own performance. In order to ensure the student network to learn the true data distribution from the teacher network, knowledge distillation with a discriminator was used for distinguishing features extracted from the teacher and student networks [Wang *et al.*, 2018a; Wang *et al.*, 2018b; Liu *et al.*, 2018]. In our method, knowledge distillation is considered as a progressive learning process, which can continuously improve the performance of student networks.

GANs [Goodfellow *et al.*, 2014] have been applied to many real world applications such as domain transfer, image generation, and novelty detection. However, to our knowledge, there is no related works that deploy the knowledge distillation on two standard GANs. Therefore, this paper designs a distillation loss to transfer knowledge from the teacher GAN to the student GAN.

2.3 The Architecture of Ganomaly

Akcay et al. [Akçay *et al.*, 2018] proposed Ganomaly for multi-class anomaly detection, in which multiple class of samples as normal data and one class of samples as abnormal data. In this paper, we utilize Ganomaly architecture for one-class novelty detection. One-class means that only the instances in one category are regarded as normal data, and the remaining categories are abnormal data.

GAN consists of two adversial modules, a generator G and a discriminator D. As is shown in the student GAN of Figure 1(a), the Ganomaly [Akçay *et al.*, 2018] framework is composed of two modules: 1) an encoder-decoder-encoder ($G_E - G_D - G_R$) pipeline based generator G that learns the distribution of input image x, where $x \in \mathbb{R}^{w \times h \times c}$, from latent spaces z_1, z_2, where $z_1, z_2 \in \mathbb{R}^d$; 2) a discriminator D that decides whether the reconstructed image $\hat{x}$ is real or fake. D and G are simultaneously optimized by playing the following minmax game as:

$$\begin{aligned}\min_G \max_D V(D,G) =& \mathbb{E}_{x\sim X}[logD(x)]+ \\ &+\mathbb{E}_{x\sim X}[log(1-G_D(G_E(x)))],\end{aligned} \tag{1}$$

where training dataset X comprises N normal images, $X = [x_1, x_2, ..., x_N] \in \mathbb{R}^{N\times w\times h\times c}$, and $\mathbb{E}_{x\sim X}$ is the expected value of x obeying distribution of normal images X.

During training, the generator loss L_g^S and the cross-entropy loss S_{ce} are minimized to train the student G and student D, respectively. The model is trained from normal samples, therefore the reconstruction error is large on abnormal samples. In the previous methods [Schlegl *et al.*, 2017; Zenati *et al.*, 2018; Perera *et al.*, 2019], the reconstruction errors of the images or latent vectors are used for anomaly detection. In Ganomaly [Akçay *et al.*, 2018], the difference S_{enc} between two latent vectors z_1, z_2 is used as novelty score, which is defined in Eq. 3.

3 Our Method

In this work, we adopt the Ganomaly [Akçay *et al.*, 2018] framework for one-class novelty detection and achieve state-of-the-art performance. To compress deep neural networks and deploy them to embedded devices with limited resources, we propose the Progressive Knowledge Distillation with GANs (P-KDGAN) to learn a lightweight student GAN from a pre-trained teacher GAN. The P-KDGAN method is composed of three modules. 1) Knowledge distillation with GANs (KDGAN), in which the distillation losses based on Ganomaly framework are proposed for transferring knowledge from the teacher GAN to the student GAN. 2) Four distillation structures are designed for KDGAN. 3) The two-step progressive learning of KDGAN can continuously improve the performance of the student GAN.

3.1 KDGAN

In the KDGAN, both the teacher GAN and the student GAN follow the network architecture of Ganomaly [Akçay *et al.*, 2018] and have the same network layers. The difference between them is the number of channels in each layer. Therefore, the generator loss L_g^T and the discriminator loss L_d^T in the teacher GAN have the same form as L_g^S and L_d^S. The generator loss L_g^S of student GAN includes reconstructed image loss S_{con}, latent space loss S_{enc} and adversarial loss S_{adv}:

$$S_{con} = \mathbb{E}_{x\sim X} \left\| x - \hat{x} \right\|_1, \tag{2}$$

$$S_{enc} = \mathbb{E}_{x\sim X} \left\| z_1 - z_2 \right\|_2, \tag{3}$$

$$S_{adv} = \mathbb{E}_{x\sim X} \left\| f(x) - f(\hat{x}) \right\|_2, \tag{4}$$

$$L_g^S = w_{con} S_{con} + w_{enc} S_{enc} + w_{adv} S_{adv}, \tag{5}$$

where $f(\cdot)$ outputs the intermediate representations of discriminator D. S_{con}, S_{enc} and S_{adv} denote the reconstruction errors of the images, latent vectors and feature maps, respectively. The weighted sum of S_{con}, S_{enc} and S_{adv} constitutes the generator loss L_g^S and is minimized to train the student generator G. The discriminator loss L_d^T consists of S_{ce}.

The designed distillation loss K_l is a novel attempt for knowledge distillation on two standard GANs. As is shown in Figure 1(a), the teacher GAN and the student GAN transfer knowledge through the intermediate layers of the generators, which includes two latent vectors and one reconstructed images. In the KDGAN, we design three losses K_1, K_x and K_2 to measure the similarity of the intermediate layers. K_1 and K_2 are the $L2$ distance of latent vectors (z_1 and $\hat{z_1}$, z_2 and $\hat{z_2}$) from the teacher GAN and student GAN. K_x is the $L1$ distance of reconstructed images ($\hat{x}$, $\widetilde{x}$). Based on the above three losses, we propose distillation loss K_l as an objective function for knowledge distillation which is the weighted sum of K_1, K_x and K_2:

$$K_l = w_1 K_1 + w_x K_x + w_2 K_2, \tag{6}$$

3.2 Distillation Structures

As is shown in Figure 1(a), the designed distillation loss K_l builds a "bridge" between the teacher and student GANs for knowledge transfer. The losses in the KDGAN consist of three parts: teacher GAN losses L_g^T, L_d^T, student GAN losses L_g^S, L_d^S, and distillation loss K_l. We define the above five loss functions as the elements of set $\mathcal{L}$:

$$\mathcal{L} = \left\{ \alpha L_g^T, \beta L_d^T, \mu L_g^S, \nu L_d^S, \lambda K_l \right\}, \tag{7}$$

where α, β, μ, ν, and $\lambda \in \{0, 1\}$ indicates whether the corresponding loss is used to train the networks.

The elements in $\mathcal{L}$ can be combined into four subsets ($\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$, $\mathcal{L}_4$) to form different distillation structures according to the following two rules. The first rule is whether the teacher GAN has fixed weights; the second rule is whether the distillation loss K_l is combined with the losses L_g^S, L_d^S to train the student GAN. Before the KDGAN, a teacher GAN is trained by its own generator loss L_g^T and discriminator loss L_d^T. The designed four distillation structures are introduced as follows:

- **KDGAN-①:** $\mathcal{L}_1$=$\{K_l\}$. Without the use of real labels, the training of student network only depends on the distillation loss K_l, which results in poor detection performance. There is no adversarial networks, and the teacher network is not updated, so its training speed is the fastest.
- **KDGAN-②:** $\mathcal{L}_2$=$\left\{L_g^S, L_d^S, K_l\right\}$. The student GAN is trained by minimizing its own losses L_g^S, L_d^S and distillation loss K_l, while the teacher GAN is not updated. The adversarial network in student GAN causes its training speed to be slightly slower than KDGAN-①.
- **KDGAN-③:** $\mathcal{L}_2$=$\left\{L_g^T, L_d^T, K_l\right\}$. The teacher GAN uses its own losses L_g^T, L_d^T to train to maintain its performance, when the training of the student GAN follows KDGAN-①. Its training speed is almost the same as that of KDGAN-②.
- **KDGAN-④:** $\mathcal{L}_2$=$\left\{L_g^T, L_d^T, L_g^S, L_d^S, K_l\right\}$. The trainings of the teacher and student GANs follow KDGAN-③ and KDGAN-②, respectively. There are two adversarial networks that need to be trained simultaneously, so the training speed is the slowest.

Algorithm 1 Progressive Knowledge Distillation with GANs

Input: Pre-trained teacher G and D, training dataset with normal instances $(x_i, y_i)_{i=1}^N$, epoch M
Output: Improved student G

```
First step
Student G and D with random initialization
for m=1 to M do
    for i=1 to N do
        Update student GAN when teacher GAN with fixed weights
    end for
end for
Second step
Download the weights of the teacher and student GANs from the previous step
for m=1 to M do
    for i=1 to N do
        Update student GAN and teacher GAN together
    end for
end for
```

3.3 Progressive Learning of KDGAN

The progressive learning of KDGAN, shown in Figure 1(b), is a two-step approach that continuously improves the performance of the student GAN and achieves better performance than the single step methods. The two-step P-KDGAN is described as follows.

P-KDGAN-I. In the first step, four distillation structures are utilized to train student network. The experimental results shown in Section 4.4 demonstrate that the performance

of student network with random initialization has a large gap compared with teacher network. Therefore, considering the detection accuracy and training time of the four distillation structures, KDGAN-② is used as the first step of P-KDGAN to enable student network to learn the basics knowledge from teacher network. In the KDGAN-②, the pre-trained teacher has already converged, so the teacher network with fixed weights is used to train the student network relying on real labels and distillation knowledge.

P-KDGAN-II. In the second step, KDGAN-③ and KDGAN-④ continue to train the teacher networks, while the student networks with basic knowledge rely on distilling knowledge to fine-training, thereby further improving accuracy and stability. The fine-learning processes in this step are named as P-KDGAN-II-②③ and P-KDGAN-II-②④. The experimental results prove that the performance of student network even exceeds the teacher network in some categories of one-class novelty detection.

The above process is illustrated in Algorithm 1.

4 Experiments

In this section, the proposed P-KDGAN is evaluated on the well-known CIFAR-10 [Krizhevsky, 2009], MNIST [LeCun and Cortes, 2005] and FMNIST [Xiao *et al.*, 2017] datasets. Following previous work [Perera *et al.*, 2019], we quantify the performance of our method using the Area Under Curve (AUC) of Receiver Operating Characteristics (ROC). The performance results are analyzed in details and are compared with state-of-the-art techniques.

All the reported results are implemented using the PyTorch framework [Paszke *et al.*, 2017] on NVIDIA TITAN 2080Ti. In the experiments, the batch size and epoch are set to 1 and 500 respectively. Adam [Kingma and Ba, 2015] is used for training with a learning rate of 0.002.

4.1 Datasets

For the three experimental datasets, the training and testing partitions remain as default. In the setup, one of the classes from training dataset is considered as normal samples for training. During testing, the remaining classes are used to represent novelty samples. For example, every experiment on the CIFAR-10 dataset is trained with 5000 samples and tested with 10,000 samples. The above experiment is repeated for all the ten categories. In addition, in order to compatible with the network architectures, all the images are resized to 32×32 by Bilinear interpolation.

4.2 Network Architectures

The Ganomaly [Akçay *et al.*, 2018] framework based on encoder-decoder-encoder ($G_E - G_D - G_R$) pipeline is used in our method. G_E, G_R in the generator and discriminator D are encoders, G_D is decoder. The encoder $E(x)$ and decoder $D(x)$ follow the DCGAN [Radford *et al.*, 2016] architecture, which have three basic layers in our model. As is shown in Table 1, the basic layers consist of: convolutional layers (deconvolutional layers), batch normalization and activation. In contrast, LeakyReLU and ReLU activations are used in encoders and decoders, except for the last layer in decoder, which uses Tanh. All the convolution filters are set to 4×4.

Layer	Units	BN	Activation	Kernel
$E(x)$				
Conv2D	64	✓	LeakyReLU	4×4
Conv2D	128	✓	LeakyReLU	4×4
Conv2D	256	✓	LeakyReLU	4×4
Conv2D	256			4×4
$D(x)$				
ConvTrans2D	256	✓	ReLU	4×4
ConvTrans2D	128	✓	ReLU	4×4
ConvTrans2D	64	✓	ReLU	4×4
ConvTrans2D	3		Tanh	4×4

Table 1: The encoder and decoder architecture for our teacher GAN, layer by layer. Units refer to number of filters in the case of convolution layers, and BN is Batch Normalization abbreviated.

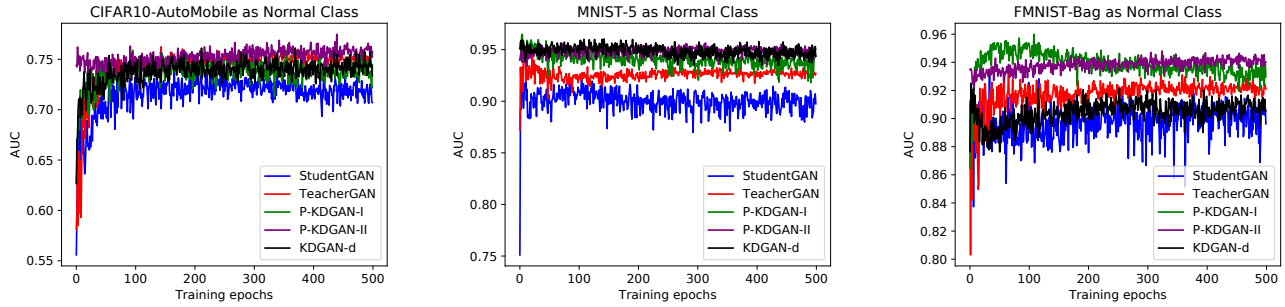
NORMAL CLASS	OCSVM	KDE	VAE	AND	AnoGAN	DSVDD	OCGAN	Ours
AIRPLANE	0.630	0.658	0.700	0.717	0.671	0.617	0.757	**0.825**
AUTOMOBILE	0.440	0.520	0.386	0.494	0.547	0.659	0.531	**0.744**
BIRD	0.649	0.657	0.679	0.662	0.529	0.508	0.640	**0.703**
CAT	0.487	0.497	0.535	0.527	0.545	0.591	0.620	**0.605**
DEER	0.735	0.727	0.748	0.736	0.651	0.609	0.723	**0.765**
DOG	0.500	0.496	0.523	0.504	0.603	**0.657**	0.620	0.652
FROG	0.725	0.758	0.687	0.726	0.585	0.677	0.723	**0.797**
HORSE	0.533	0.564	0.493	0.560	0.625	0.673	0.575	**0.723**
SHIP	0.649	0.680	0.696	0.680	0.758	0.759	0.820	**0.827**
TRUCK	0.508	0.540	0.386	0.566	0.665	0.731	0.554	**0.735**
MEAN	0.5856	0.6097	0.5833	0.6172	0.6179	0.6481	0.6566	**0.7376**
0	0.988	0.885	0.997	0.984	0.966	0.980	**0.998**	0.996
1	0.999	0.996	0.999	0.995	0.992	0.997	**0.999**	**0.999**
2	0.902	0.710	0.936	0.947	0.850	0.917	0.942	**0.969**
3	0.950	0.693	0.959	0.952	0.887	0.919	0.963	**0.969**
4	0.955	0.844	0.973	0.960	0.894	0.949	**0.975**	0.970
5	0.968	0.776	0.964	0.971	0.883	0.885	**0.980**	0.951
6	0.978	0.861	0.993	0.991	0.947	0.983	0.991	**0.992**
7	0.965	0.884	0.976	0.970	0.935	0.946	0.981	**0.982**
8	0.853	0.669	0.923	0.922	0.849	0.939	0.939	**0.965**
9	0.955	0.825	0.976	0.979	0.924	0.965	0.981	**0.987**
MEAN	0.9513	0.8143	0.9696	0.9671	9127	0.9480	0.9750	**0.9780**

Table 2: One-class novelty detection results on CIFAR-10 and MNIST dataset. The average AUC of three repeated experiments was used as detection performance.

The difference between a teacher network and a student network is the number of channels in the intermediate representations. For the three experimental datasets, the intermediate layers in the teacher networks are set to 64-128-256 channels following the OCGAN [Perera *et al.*, 2019]. The student networks in each dataset utilize intermediate representations with 8-16-64 channels, 2-4-8 channels and 1-2-4 channels respectively. The encoder $E(x)$ and decoder $D(x)$ architecture of the teacher GAN is illustrated in Table 1.

4.3 Results on One-class Novelty Detection

In this section, we compare our Ganomaly [Akçay *et al.*, 2018] based Teacher GAN with several traditional and deep learning based methods on CIFAR-10 and MNIST datasets , including one-class SVM (OC-SVM) [Schölkopf *et al.*, 2001], kernel density estimation (KDE) [Parzen, 1962], deep variational autoencoder (VAE) [Kingma and Welling, 2014], AND [Abati *et al.*, 2019], AnoGAN [Schlegl *et al.*, 2017], DSVDD [Ruff *et al.*, 2018] and OCGAN [Perera *et al.*, 2019]. In light of massive experiments, the parameters of w_{con}, w_{enc} and w_{adv} in Eq. 5 are mannually configured as 10, 1 and 1. The parameters of w_1, w_x and w_2 in Eq. 6 are set as 1. We take the average AUC of the last epoch from multiple trials, but not the manually selected result, as the detection performance, which is more convictive.



(a) (b) (c)

Figure 2: Training curves of the AUC on three datasets. The normal classes are: (a) AutoMobile on CIFAR-10. (b) 5 on MNIST. (c) Bag on FMNIST. P-KDGAN-I represents KDGAN-②. P-KDGAN-II represents P-KDGAN-II-②③. KDGAN-d represents KDGAN-④.

Method	CIFAR-10	MNIST	FMNIST
KDGAN-①	71.59%	96.7%	92.48%
KDGAN-②	72.43%	96.75%	92.41%
KDGAN-③	70.94%	97.18%	92.90%
KDGAN-④	72.58%	96.87%	92.42%
P-KDGAN-II-②③	73.05%	97.25%	92.93%
P-KDGAN-II-②④	72.52%	96.67%	92.57%

Table 3: Compare the performance of KDGAN and P-KDGAN. We highlight the best results in red and the second-best results in blue color.

Comparisons on CIFAR-10 and MNIST. The performance of one-class novelty detection on CIFAR-10 dataset, our method shown in Table 2 achieves 73.76%, which is higher than the best OCGAN [Perera *et al.*, 2019] method about 8%. For MNIST dataset, our method achieves 97.80% yielding an improvement of about 0.3% compared with state-of-the-art method.

4.4 Evaluation of P-KDGAN Method

In this section, the progressive knowledge distillation with GANs is evaluated on CIFAR-10, MNIST and FMNIST datasets. In each experiment, the weights of the last epoch are served as the teacher network.

KDGAN vs. P-KDGAN. As is shown in Table 3, P-KDGAN-II-②③ achieves the best performance on three datasets, which illustrates the effectiveness of our progressive learning of KDGAN. Although KDGAN-③ achieves the second-best results on MNIST and FMNIST, it shows the worst performance on CIFAR-10 dataset. KDGAN-④ obtain the second-best results on CIFAR-10, but it was about 0.5% lower than the best result. In addition, KDGAN-d (KDGAN-④) iillustrated in Figure 2(c) is inferior in accuracy and training stability compared to P-KDGAN-II. The training curves of the AUC illustrated in Figure 2 clearly shows that proposed P-KDGAN-II can improve the accuracy of the student network and even surpass the teacher network, and reduce shock. Therefore, the above analysis concludes that student networks with random initialization can only learn the basic knowledge of the teacher networks, and the fine-training in the second step of P-KDGAN can further improve performance.

Dataset	Method	AUC. ↓	#Param. ↓	#FLOPs. ↓
CIFAR-10	Teacher	73.76%	5.12M	56M
	Student	3.15%	6.22×	24.45×
	P-KDGAN	**0.71%**		
MNIST	Teacher	97.80%	5.12M	56M
	Student	2.32%	52.22×	311.11×
	P-KDGAN	**0.55%**		
FMNIST	Teacher	93.11%	5.12M	56M
	Student	1.91%	105.45×	700×
	P-KDGAN	**0.18%**		

Table 4: Evaluation of our P-KDGAN method on CIFAR-10, MNIST and FMNIST datasets. (M means million, # means the compression ratio of parameter numbers and FLOPs compared to the teacher GAN.)

Results on P-KDGAN. As is illustrated in Table 4, the performance of the student GAN obtained by two-step P-KDGAN is only 0.71%, 0.55% and 0.18% lower than that of the teacher GAN when compressing the computation at ratios of 24.45:1, 311.11:1, and 700:1, respectively.

5 Conclusion

In this paper, we use the encoder-decoder-encoder pipeline based GANs for one-class novelty detection and achieve state-of-the-art performance. To compress the model, the progressive knowledge distillation with GANs is proposed, which is a novel exploration that applies the knowledge distillation on two standard GANs. The two-step progressive learning can continuously improve the performance and reduce shock of the student network, in which the designed distillation loss plays an important role. Experiments on three datasets validate the effectiveness of our proposed method. Moreover, our proposed method can be used to compress other GANs-based applications, such as image generation.

Acknowledgments

This work is supported by Key-Area Research and Development Program of Guangdong Province (2019B010155003), National Natural Science Foundation of China (U1713203), Shenzhen Science and Technology Innovation Commission (Project KQJSCX20180330170238897), and the Scientific Instrument Developing Project of the Chinese Academy of Sciences (Grant No. YJKYYQ20190028).

References

[Abati *et al.*, 2019] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. *CVPR*, pages 481–490, 2019.

[Akçay *et al.*, 2018] Samet Akçay, Amir Atapour Abarghouei, and Toby P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. *ACCV*, 2018.

[Chandola *et al.*, 2009] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection : A survey. *ACM Comput. Surv.*, 41(3):1–72, 2009.

[Cheng *et al.*, 2017] Y. Cheng, D. Wang, P. Zhou, and T. Zhang. A survey of model compression and acceleration for deep neural networks. *ArXiv*, abs/1710.09282, 2017.

[Donahue *et al.*, 2017] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *ICLR*, 2017.

[Erfani *et al.*, 2016] Sarah M. Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

[García-Teodoro *et al.*, 2009] Pedro García-Teodoro, Jesús E. Díaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput. Secur.*, 28(1-2):18–28, 2009.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014.

[Hinton *et al.*, 2015] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[LeCun and Cortes, 2005] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. 2005.

[Liu *et al.*, 2018] Peiye Liu, Wu Liu, Huadong Ma, Tao Mei, and Mingoo Seok. Ktan: knowledge transfer adversarial network. *ArXiv*, abs/1810.08126, 2018.

[Parzen, 1962] Emanuel Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. *NeurIPS*, 2017.

[Perera *et al.*, 2019] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. *CVPR*, pages 2893–2901, 2019.

[Radford *et al.*, 2016] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016.

[Romero *et al.*, 2015] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *ICLR*, 2015.

[Ruff *et al.*, 2018] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert A. Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. *ICML*, 2018.

[Schlegl *et al.*, 2017] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *IPMI*, 2017.

[Schölkopf *et al.*, 2001] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alexander J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[Srivastava *et al.*, 2008] Abhinav Srivastava, Amlan Kundu, Shamik Sural, and Arun K. Majumdar. Credit card fraud detection using hidden markov model. *IEEE Transactions on Dependable and Secure Computing*, 5(1):37–48, 2008.

[Vincent *et al.*, 2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. *ICML*, 2008.

[Wang *et al.*, 2018a] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Kdgan: knowledge distillation with generative adversarial networks. *NeurIPS*, 2018.

[Wang *et al.*, 2018b] Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Adversarial learning of portable student networks. *AAAI*, 2018.

[Wold *et al.*, 1987] Svante Wold, Kim H. Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52, 1987.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: A novel image dataset for benchmarking machine learning. *ArXiv*, abs/1708.07747, 2017.

[Zenati *et al.*, 2018] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *ArXiv*, abs/1802.06222, 2018.