CNN based Metrics for Performance Evaluation of Generative Adversarial Networks

Adarsh Prasad Behera, Satya Prakash, Siddhant Khanna, Shivangi Nigam, and Shekhar Verma, *Senior member, IEEE*

***Abstract*—In this work, we propose two Convolutional Neural Network (CNN) based metrics, Classification Score (CS) and Distribution Score (DS), for performance evaluation of Generative Adversarial Networks (GANs). Though GAN-generated images can be evaluated through manual assessment of visual fidelity, it is prolonged, subjective, challenging, tiresome, and can be misleading. Existing quantitative methods are biased towards memory GAN and fail to detect over-fitting. CS and DS allow us to experimentally prove that training of GANs is actually guided by the data set, that it improves with every epoch and gets closer to following the distribution of the data set. Both methods are based on GAN-generated image classification by CNN. CS is the root mean square (RMS) value of three different classification techniques, Direct Classification (DC), Indirect Classification (IC), and Blind Classification (BC). It exhibits the degree to which GAN can learn the features and generate fake images similar to real data sets. DS shows the contrast between the mean distribution of GAN-generated data and the real data. It indicates the extent to which GANs can create synthetic images with similar distribution to real data sets. We evaluated CS and DS metrics for different variants of GANs and compared their performances with existing metrics. Results show that CS and DS can evaluate the different variants of GANs quantitatively and qualitatively while detecting over-fitting and mode collapse.**

***Impact Statement*—This research marks a pivotal advancement by addressing critical flaws in Generative Adversarial Networks (GANs), offering enhanced reliability in assessing model performance. Its breakthrough in detecting over-fitting and mode collapse ensures greater trustworthiness and precision, fostering advancements in robust GAN development and real-world applications. To the best of our knowledge, none of the preceding studies have demonstrated similar capabilities.**

***Index Terms*—CNN, classification score, distribution score, GANs, generative models, quantitative performance evaluation, image classification.**

I. Introduction

ADVERSARIAL process enables us to learn implicit generative models known as GANs[1]. GANs can be trained through both semi-supervised and unsupervised learning. The fundamental idea of a GAN model is based on a two-person min-max zero-sum game. There are two different networks, the generator, and the discriminator, in a GAN model that correspond to the game's two players. These players compete with one another. Real data is not accessible to the generator, and it tries to generate fake data similar to real data from noise. Both real and fake data are accessible to the discriminator, which distinguishes between the two and gives feedback to the generator to learn the features and distribution of real images [2]. Both the models are trained simultaneously until the discriminator cannot distinguish data produced by the generator and real data. These two networks are usually convolutional networks or fully connected layers [3].

In addition to realistic image synthesizing [4], GANs have been used in applications such as dialogue generation [5], simulated image refinement [6], semantic segmentation [7], object detection [8], text modeling [9], image-to-image translation [10], image captioning [11]–[13] and in-painting [14], [15]. In [16], Defence-GAN is proposed that shield against different adversarial attacks such as black-box and white-box attacks. Despite having abundant application and availability of GAN models, performance evaluation of such models is still qualitative. Manual assessment of the visual fidelity of synthesized images is used to measure performance. This is prolonged, subjective, challenging, tiresome, and often misleading.

In GANs, the generator and discriminator accuracy demonstrate their performance relative to each other, which is not an appropriate metric to measure the quality and diversity of the generated images. Various evaluation metrics have been defined for performance evaluation of GANs. Some of the qualitative methods like Nearest neighbour [17], rating and preference judgment [18], and rapid scene categorization [19] have been proposed to measure the performance of GANs based on the quality of generated images. However, these methods are largely biased towards over-fitting. In [20], a Parzen window density metric was proposed in which a Parzen window mix is constructed by taking some samples of synthesized images generated by a GAN and using these samples as centroids of a Gaussian mixture. The log-likelihood score is calculated on test data to evaluate the performance. However, in high dimensional space, the presumption of the Gaussian model may not be able to predict memory GAN and GAN over-fitting. In Inception Score (IS) [21], a pre-trained neural network store preferable attributes of GAN-generated images, such as high diversity and classifiability. Though IS gives a fair correlation with the diversity and quality of GAN-generated samples, it cannot detect over-fitting as it is biased towards memory GAN. It remembers the real image set and generates only the identical samples [4]. It also fails to detect "Mode collapse." Frechet Inception Distance (FID) was proposed to improve IS in [22]. In FID, a sample of synthesized images is embedded in a feature space defined by a certain Inception Net layer. The mean and co-variance

Adarsh Prasad Behera is with the Edge Networks Group at IMDEA Networks Institute, Leganés, Madrid, 28918 Spain, e-mail: adarsh.behera@imdea.org

Satya Prakash, Siddhant Khanna, Shivangi Nigam, and Shekhar Verma, are with the Department of Information Technology, Indian Institute of Information Technology, Allahabad, U.P. , 211012 India, e-mails: mit2019111@iiita.ac.in, pro.siddhant@iiita.ac.in, rsi2018506@iiita.ac.in, sverma@iiita.ac.in

are calculated for both real and generated data assuming the embedded layer is a continuous multivariate Gaussian. The major drawback of FID is the assumption that the features are always Gaussian.

Image quality measures like SSIM and peak-signal-to-noise ratio (PSNR) are used for training and evaluation of GANs[23]. In PSNR, the quality of two monochrome images, A and B, is compared through the signal-to-noise ratio. A higher value of PSNR indicates better quality of the image. The quality of the synthesized image is compared to the quality of the real image through PSNR. The major drawback of PSNR is the requirement of an appropriate reference image for each synthesized image. Maximum Mean Discrepancy (MMD) [24] draws samples independently from real and synthetic data sets and computes the difference between the probability distribution of both samples. It is highly biased towards memory GANs and cannot detect over-fitting. A comparative performance analysis metric, Generative Adversarial Metric (GAM), as defined in [25] in which two GANs compete against one another by exchanging generators or discriminators across each other. The likelihood ratio of both models gives an idea about the relative performance. However, the two models must have similar performing discriminators, which is not feasible in practice. The computational cost of this metric is also very high. In [26], a metric was proposed to compute precision to measure the quality of the synthesized images and recall to measure the distribution proportion of real and generated data. However, these scores are only practical for synthetic data and not be used to compute in real-life data sets.

Though several metrics have been developed for the performance evaluation of GANs, no metric has been accepted in consensus. In this work, we define novel metrics using image classification through CNN to overcome the existing limitations. The major contributions of our work are

- Two novel metrics based on image classification through CNN to evaluate the performance of GANs.
- A novel method to detect over-fitting in GANs using image classification.
- A novel method to detect mode collapse in GANs using image classification.
- Supervise GAN models using the proposed methods for both feature and distribution learning.

The rest of the paper is structured under different sections as follows. In section II, GAN and its different variants are discussed in detail. In section III, a brief overview of the problem definition is presented. The system model is presented in section IV. Section V explains our proposed methodology along with all the required algorithms to implement it. Section VI contains the result analysis; at the end, section VII concludes the paper.

II. GAN and its Different Variants

GAN model consists of a pair of deep neural nets termed the discriminator (D) and the generator (G). The generative model takes input given as random uniform noise and tries to generate fake images identical to real images. In contrast, the discriminative model takes input from real and fake images and tries to distinguish between them. Feedback received by the generator from the discriminator, when it evaluates the forged images, helps the generator perform better by fine-tuning. The whole structure depicts a min-max game with two opponents in which the generator tries to minimize its loss by optimizing an objective function. In contrast, the discriminator tries to maximize it, as shown in $fig.1$. The ultimate intention of this game is given as follows:

$$\min_{Gen}\max_{Disc} V(Disc;Gen) = E_{x\ \ p_{data}(x)}[\log Disc(x)] + E_{z\ \ p_z(z)}[\log(1\quad Disc(Gen(z)))] \tag{1}$$

We consider real distribution $P_{data}(x)$ where x is modelled from this distribution while $P_z(z)$ is generated distribution over input noise z. The generator learns through a differentiable function $Gen(z;\ g)$, given input z, parameterized over g. The learning in the discriminator happens through a differentiable function $D(x;\ d)$, given input x, parameterized over d. $Disc(x)$ provides probabilistic estimate of x is derived from $P_{data}(x)$ instead of noise distribution P_z. Discriminator tries to maximize $\log Disc(x)$. On the contrary, the generator tries to minimize $\log(1\quad Disc(Gen(z)))$.

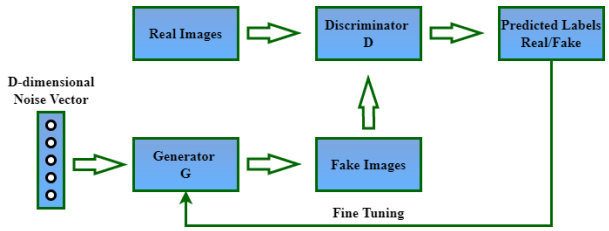

Fig. 1. Basic GAN model

With the huge popularity of GANs for image synthesis, many variants have been developed over the years. Some of the noteworthy developed models are listed below.

A. Conditional GAN (CGAN)

CGANs are GANs in which both the generator and discriminator are conditioned with some additional information y. This additional information y could be class labels and is added to overcome the limitations of the original model due to the over-dependence on random variables [26]. In the discriminator network, the additional information y along with real images x are input to the function $Disc$. So, now the min-max equation becomes

$$\min_{Gen}\max_{Disc} V(Disc;Gen) = E_{x\ \ p_{data}(x)}[\log Disc(xjy)] + E_{z\ \ p_z(z)}[\log(1\quad Disc(Gen(zjy)))] \tag{2}$$

B. Deep Convolutional GAN (DCGAN)

The core of the DCGANs structure was to use CNN architecture in Discriminator and Generator models [27]. CNNs are modified to adopt the following constraints.

Stridden convolutions replace fully-connected layers, pooling layers, in discriminator and are useful in down-sampling images, while transposed convolutions are used in up-sampling images in the generator.
Batch normalization applied to all hidden layers in discriminator and generator models helps in better gradient flow.
Usage of ReLU activation at all but the last layer of the Generator and Leaky ReLU activation function at every layer of the discriminator.

C. Information Maximizing GAN (InfoGAN)

An information-theoretic development of generative adversarial networks is proposed in [28] that can learn extracted attributes with unsupervised learning. It maximizes the mutual information among a tiny subset of the hidden variables and observations, thus called information maximizing GANs. The min-max equation of InfoGAN is

$$\min_{Gen} \max_{Disc} V_I(Disc; Gen) = V(Disc; Gen) \quad I(y; Gen(z; y)) \tag{3}$$

.

D. Adversarial Auto Encoders (AAE)

A combined adversarial auto-encoders and GANs framework is proposed in [29]. It performs variational inference in adversarial training tests by matching a prior distribution to the cumulative posterior distribution of the hidden code vector. It can be represented mathematically as

$$Q(z) = \int_x Q(z|x) P_d(x) dx \tag{4}$$

where x and z represent an input and hidden code vector, respectively. $P_d(x)$, $P(z)$ and $Q(z)$ are data distribution, prior and posterior distribution respectively. $Q(z|x)$ represents encoding distribution and regularization is done by matching $Q(z)$ to $P(z)$.

E. Wasserstein GAN (WGAN)

Wasserstein Generative Adversarial Networks (WGAN) was proposed in [30] in which Wasserstein distance or Earth-mover (EM) distance is used as a loss function instead of JS-divergence. EM is the minimum cost to transfer model data distribution P_r to target data distribution P_g. Wasserstein distance between both distributions is

$$W(P_r; P_g) = inf_{(\ 2 \ (P_r; P_g))} E_{(x;y) \ \ y}[|x \quad y|] \tag{5}$$

where $(x; y)$ calculates the amount of mass to be transformed from x to y and $(P_r; P_g)$ represents joint distributions.

III. Problem Description

Several metrics have been developed over the past few years for performance analysis of GANs, but they have yet to garner consensus as the method for performance evaluation. Qualitative methods like rapid scene categorization and nearest neighbours are biased towards over-fitting models. Quantitative methods like IS fail to detect overfitting as well as mode collapse. FID is highly biased towards memory GANs and only considers Gaussian distribution for features which is not guaranteed all the time. Image quality measure like PSNR needs an appropriate reference image and works well in monochrome images but fails to replicate such results in multi-chrome images.

The primary objective is to develop a metric that simultaneously assesses visual fidelity and diversity. Visual fidelity implies that the synthetic data set with high likelihood is generated from GAN, and diversity ensures that all modes are considered. The exigency of such a method is immense for impartial model comparison, understanding and interpreting different variants of GANs, and improvement and development of novel generative models. We propose two novel metrics based on image classification through CNN for performance evaluation of generative models to conduct an unbiased comparison of different variants of GAN.

IV. System Model

We consider a given real-life data set $S_n(S_t; S_v)$, where n is the number of different classes, and S_t and S_v represent training data and test data or validation data, respectively. Usually, the number of test data is predefined in a data set that lies somewhere around 10 20% of the total data. S can be randomly divided into two parts $S_t{}'$ and $S_v{}'$ to create training and test data with different distributions such that

$$|S_t{}'| = |S_t| \text{ and } |S_v{}'| = |S_v|$$

The GANs are trained using either S_t or $S_t{}'$, and the generated synthetic data sets can be termed as S_g and $S_g{}'$, respectively. As the distributions of real data set S_t and $S_t{}'$ are different, the synthetic data sets S_g and $S_g{}'$ will follow the distribution of their corresponding real data sets and have different distributions as well.

The distribution of each class's images can be represented by their normalized fractional value and represented by $P_{X\,i}$, where X is the respective data set and i is the respective class. The cumulative value of all the P_is will always be equal to 1. For example, the cumulative value of distributions of predefined training data can be represented as

$$P_t = \sum_{i=1}^{X} P_{t\,i} = 1 \tag{6}$$

V. Proposed Evaluation Methodology

A metric to evaluate GAN must be able to determine whether, a GAN is able to generate images that are realistic and, also, has the ability to generate all possible realistic images. Our objective is to develop a metric that can achieve this by evaluating GANs based on their performance and prediction

of over-fitting and mode collapse. Image classification through CNN is used as the basic principle of the evaluation. First, we train a CNN $C1$ with the predefined training data S_t and check its accuracy for predefined test data S_v. Then, S_t and S_v are merged together and randomly divided into $S_t{}^0$ and $S_v{}^0$. GAN is trained using new training data $S_t{}^0$, and pre-trained $C1$ predicts the labels of the generated images. With the help of these predicted labels of $S_g{}^0$, we define different image classifications and compute accuracies to evaluate the performance of the variants of GANs. An illustration of the system model is shown in *fig:*2.

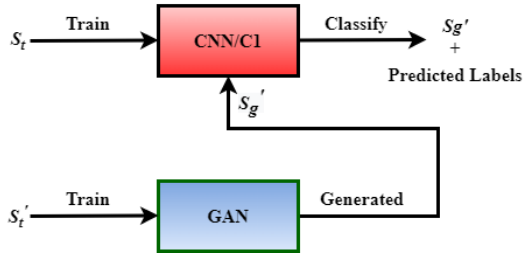

Fig. 2. System model

A. Classification Score

CS indicates the extent to which a GAN has learnt the features of the real images and whether the synthetic images have similar features to that of real images. CS can be determined by three different classification methods: Direct Classification (DC), Indirect Classification (IC), and Blind Classification (BC).

1) Direct Classification: In this, an image classifier or CNN $C2$ is trained with the same training data as the GAN $S_t{}^0$ and the accuracy is tested with $S_g{}^0$ as test data. A decent-performing GAN should be able to generate good accuracy in DC. It represents the "precision' measure of the corresponding GAN.

2) Indirect Classification: In this, a CNN $C3$ is trained with GAN-generated synthetic data $S_g{}^0$, and the accuracy is tested with $S_v{}^0$ as test data. As $C3$ is trained through synthetic images produced by GAN, the accuracy should be higher once the GAN converges. Higher accuracy of IC means GAN has learned well and has generated images with similar features to training data. However, if both DC and IC accuracies are too high, it means the GAN has just memorized the training data set and is not producing any new or different data. The RMS value of DC and IC can be compared with the accuracy of $C1$. This can be used as a threshold value for detecting over-fitting.

3) Blind Classification: In this, a CNN $C4$ is trained with $S_v{}^0$ as training data, and the accuracy is tested with $S_g{}^0$ as test data. As data are randomly divided into training and test data sets, $S_v{}^0$ will contain different images with similar features to that of $S_t{}^0$. Higher accuracy in BC means GAN has learned the features quite well with good results. An illustration of DC, IC, and BC is shown in *fig:*3.

It can be observed that, for a decent-performing GAN, all three accuracies will be on the higher side. For comparative

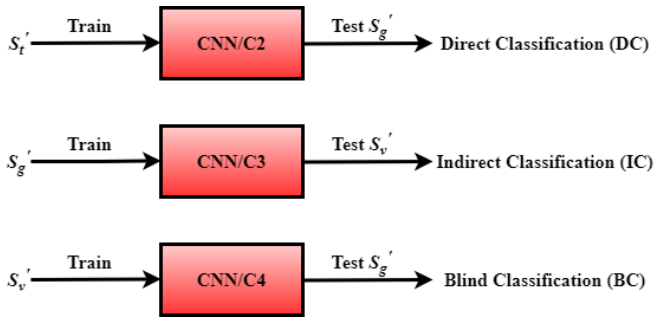

Fig. 3. Direct, Indirect and Blind classifications

performance analysis, we need to differentiate GANs based on their performances in each of the three classification accuracy. We consider the RMS value of all three classifications to define CS so that higher values will have higher weightage and vice versa. This ensures an unbiased evaluation of GANs even if any GAN fails to perform well in any one of the classification accuracies.

$$CS = \sqrt{\frac{DC^2 + IC^2 + BC^2}{3}} \tag{7}$$

A higher value of CS represents better learning of GAN and the generation of high-quality synthetic images.

Algorithm 1 Accuracy Calculation

procedure A=ACCURACY($S_t; S_v$)
  Train the CNN with S_t
  Predict the labels of S_v from $C1$
  Match the predicted labels with original labels
  Compute $A = \frac{|\text{correctly predicted labels}|}{j S_v j}$
  return A
end procedure

Algorithm 2 Classification Score

procedure CS=CLASSIFICATION SCORE(S_n)
  $C1$=ACCURACY($S_t; S_v$)
  Randomly divide S_n into $S_t{}^0$ and $S_v{}^0$
  Train GAN with $S_t{}^0$ and generate $S_g{}^0$
  Predict the labels of $S_g{}^0$ from $C1$
  DC=ACCURACY($S_t{}^0; S_g{}^0$)
  IC=$\sqrt{\text{ACCURACY}(S_g{}^0; S_v{}^0)}$
  if $\frac{DC^2 + IC^2}{2}$ $C1$ **then**
    Over-fitting of GAN is occurring
  end if
  BC=$\sqrt{\text{ACCURACY}(S_v{}^0; S_g{}^0)}$
  Compute $CS = \sqrt{\frac{DC^2 + IC^2 + BC^2}{3}}$
  return CS
end procedure

B. Distribution Score

DS indicates the extent to which GANs can create synthetic images with a distribution similar to that of real data sets.

Since training and test data are chosen and separated randomly, the distribution of S_t^0 must differ from that of predefined S_t. A necessary but insufficient condition for a well-performing GAN is that the synthetic image set S_g^0 should follow the distribution of S_t^0 closely. The minimum the contrast, the better GAN has learned. To compute the mean contrast in the distribution of the synthetic data set from the real data set, the normalized fractional value of the distribution of each class is required. So, after completing GAN training and generation of S_g^0, the labels are predicted by pre-trained CNN $C1$. After that, each class's distribution's individual normalized fractional value is calculated for S_g^0 and S_t^0. Then the contrast in the distribution of each class is calculated, and finally, the RMS value of all the contrasts is subtracted from 1 to yield the DS. So, DS can be expressed as

$$DS = 1 - \sqrt{\sum_{i=1}^{n} \frac{(P_{g_i}^0 - P_{t_i}^0)^2}{n}} \tag{8}$$

where $P_{g_i}^0$ and $P_{t_i}^0$ represent the normalised fractional distributions of respective class i of S_g^0 and S_t^0 respectively. A higher value of DS indicates that GAN has learned the distribution of real data sets well enough and produced a similar distribution.

Furthermore, DS can be used to detect mode collapse in GANs. In mode collapse, GAN fails to learn the distribution and generates a limited variety of samples. This will result in a lower DS. In a single mode collapse, the DS value will always be lesser than the normalized fractional value of the distribution of the mode or class with the maximum number of samples in the training data set ($P_{t_{max}}^0$). This can be used as a threshold value for detecting mode collapse.

Algorithm 3 Distribution Score

procedure DS=DISTRIBUTION SCORE(S_n)
  $C1$=ACCURACY($S_t; S_v$)
  Randomly divide S_n into S_t^0 and S_v^0
  Train GAN with S_t^0 and generate S_g^0
  Predict the labels of S_g^0 from $C1$
  Compute $P_{t_i}^0$ and $P_{g_i}^0$
  Compute $DS = 1 - \sqrt{\sum_{i=1}^{n} \frac{(P_{g_i}' - P_{t_i}')^2}{n}}$
  if $DS < P_{t_{max}}^0$ **then**
    Mode collapse is occurring
  end if
  return DS
end procedure

CS evaluates the GAN's ability to replicate real dataset features by employing three different classification techniques: DC, IC, and BC. Unlike traditional metrics, CS provides a comprehensive assessment of the GAN's capability to learn and generate images that closely resemble real data. This multi-faceted approach allows for a more nuanced understanding of the model's performance and its potential limitations, particularly in detecting overfitting.

On the other hand, DS focuses on measuring the disparity between the mean distribution of GAN-generated data and the real data. By quantifying this difference, DS provides valuable insights into how well the GAN can create synthetic images with distributions similar to real datasets. This unique aspect of DS enables researchers to identify and address issues such as mode collapse, where the GAN fails to capture the entire diversity of the underlying data distribution. Additionally, CS and DS leverages pre-trained CNNs, ensuring robustness and reliability in evaluating GAN-generated images.

VI. Results And Discussion

A. Data sets

We evaluate the performance of various GANs on three different data sets MNIST, Fashion-MNIST and CIFAR-10. MNIST is a large black-and-white image collection of handwritten digits. It has 10 different classes, each one for a different digit ranging from 0 9. It has 70000 images split into 60000 and 10000 images to be used for training and testing purposes, respectively. The image size for MNIST is 28 28 pixels.

Fashion-MNIST (FMNIST) is a similar large-scale black-and-white image collection of different fashion items. It has 10 different classes such as 0: T-shirt/top, 1: Trouser, 2: Pullover, 3: Dress, 4: Coat, 5: Sandal, 6: Shirt, 7: Sneaker, 8: Bag, 9: Ankle boot. Like MNIST, these are grey-scale images of size 28 28 pixels and 70000 images split into 60000 and 10000 images to be used for training and testing purposes, respectively.

CIFAR-10 is a colour image data set of 10 different classes representing different vehicles, animals and birds. The 10 different classes represent aeroplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 60000 low-resolution colour images of size 32 32 pixels. Each class has 6000 different images, and the data set can be split into 50000 and 10000 images to be used for training and testing purposes, respectively.

The respective neural network architectures used for image classification on MNIST, FMNIST and CIFAR-10 are shown in *fig:* 4, 5, and 6 respectively. *Table* I shows the hyperparameters and training configurations used for different datasets.

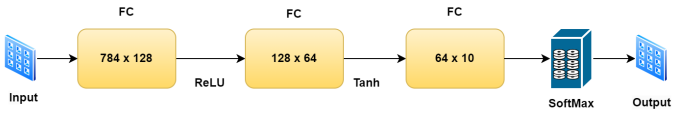

Fig. 4. Neural Network architecture used for MNIST dataset

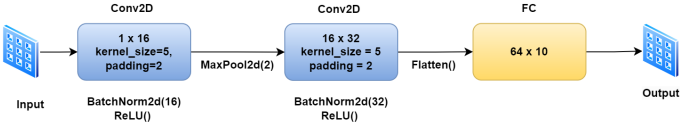

Fig. 5. Neural Network architecture used for Fashion-MNIST dataset

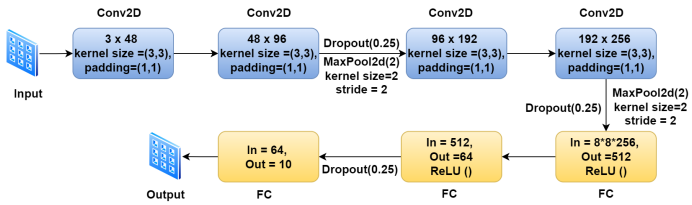

Fig. 6. Neural Network architecture used for CIFAR-10 dataset

TABLE I
HYPERPARAMETERS AND TRAINING CONFIGURATIONS FOR THE EXPERIMENTAL SETUP

Hyperparameters	MNIST	F-MNIST	CIFAR-10
batch size	64	64	50
learning rate	0.01	0.001	0.001
Epochs	15	5	25
Optimizer	SGD	Adam	Adam

B. Performance Analysis

We considered five different variants of GANs: CGAN, DCGAN, InfoGAN, AAE, and WGAN for comparative performance analysis. First, the data set is trained on a CNN using S_t and $C1$ is calculated. After that, random splitting is done, and different GAN variants are trained using S_t^0 for 200 epochs for MNIST and F-MNIST datasets, and for 1000 epochs for CIFAR-10 dataset. Subsequently, CS is calculated through DC, IC, and BC. For the computation of DS, we split the entire data set with varied distributions to oversee the learning of GANs through different epochs. We randomly chose 5000 sample images from each of the first five classes. Similarly, 3000 sample images are chosen randomly from each of the last five classes to form the training set S_t^0. Four state-of-the-art quantitative evaluation metrics: FID, IS, PSNR, and MMD, are also computed for different epochs. Finally, all the metrics are analyzed, and comparative performance evaluation is done for both data sets.

1) MNIST: A three-layer neural network is trained for 15 epochs with a learning rate of 0:01 and used for image classification in C1, C2, C3, and C4. The accuracy of C1 is 0:9818, termed 'validation accuracy' for future references and the classified images along with their respective classes are shown in *fig:*8. CS and DS, along with FID, IS, PSNR, and MMD, are calculated for all the variants of GANs. It is observed in *Table* II and *fig:*7 that although CGAN performs marginally better in DC and BC, AAE outperforms all the variants regarding overall classification score and all other state-of-the-art evaluation metrics. It validates the earlier findings that AAE obtains the state-of-the-art classification results [7]. The reason behind the better performance of CGAN in the case of DC and BC is the labels associated with the images being generated. In *fig:*7, it can be observed that for initial epochs, the CS value is comparatively low for each variant, and it gradually increases with an increase in the number of epochs. Once the GAN converges, the graph becomes stagnant except for InfoGAN, in which case the accuracy starts to decrease after the point of convergence. CS performance of AAE is followed by CGAN, DCGAN, InfoGAN, and WGAN, respectively. An example of memory GAN was also tested for over-fitting, yielding a $C2$ accuracy of 0:9932 and $C3$ accuracy of 0:9769, respectively. The RMS value of $C2$ and $C3$ accuracies is 0:9850, which is more than the threshold value of validation accuracy 0:9818, proving the occurrence of over-fitting.

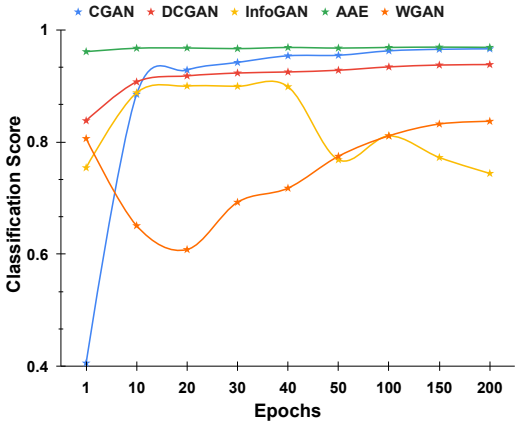

Fig. 7. Classification Score Vs Number of Epochs for MNIST

For DS, we considered different distributions for training as each class has an almost similar number of samples in the original training data set. It can be observed in *fig:*9 that after some initial number of epochs, each of the GAN variants performs reasonably well and learns the real data distribution to a large extent. DCGAN outperforms the rest of the variants with a DS of 0:9960, closely followed by AAE, WGAN, InfoGAN, and CGAN, respectively.

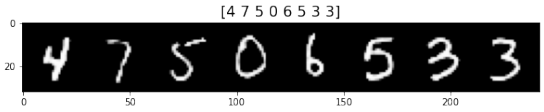

Fig. 8. MNIST images generated from AAE and subsequent labeling done by $C1$

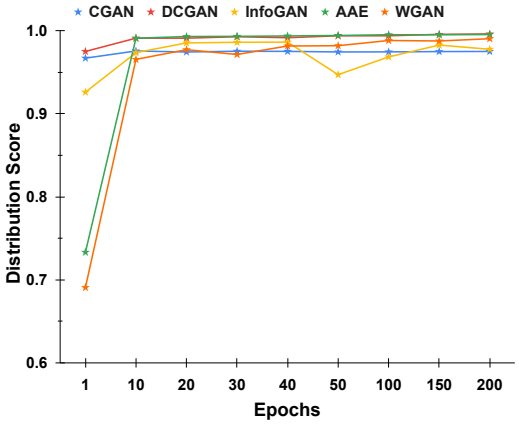

Fig. 9. Distribution Score Vs Number of Epochs for MNIST

2) Fashion-MNIST (F-MNIST): A three-layer CNN is trained for 5 epochs with a learning rate of 0:001 and used

TABLE II
PERFORMANCE EVALUATION OF VARIOUS GANS ON MNIST DATA SET

MNIST (Validation accuracy from C1=0.9818)										
Metrics ! **GANs #**	DC	IC	BC	**CS**	**DS**	**FID**	**IS**	**PSNR**	**MMD**	**Mode collapse or Over-fitting**
CGAN	**0.9844**	0.9524	**0.9620**	0.9664	0.9749	65.82	1.85	51.16	0.1249	None
DCGAN	0.9386	0.9697	0.9056	0.9383	**0.9960**	53.01	2.01	50.71	0.1247	None
InfoGAN	0.8828	0.9488	0.8624	0.8987	0.9861	71.93	1.83	56.97	0.1250	None
AAE	0.9836	**0.9745**	0.9499	**0.9695**	0.9950	**24.26**	**2.05**	**65.68**	**0.1183**	None
WGAN	0.7915	0.9576	0.7473	0.8370	0.9902	125.80	1.82	51.15	0.1249	None

for image classification in C1, C2, C3, and C4. The validation accuracy for F-MNIST is 0:9030 and the classified images along with their respective classes are shown in *fig:*11. All the variants of GANs are trained for 200 epochs, and the value at the point of convergence (or the best performance) is considered for comparative analysis. CS, DS FID, IS, PSNR, and MMD are calculated. It can be observed from *Table* III and *fig:*10 that, initially, CS for each variant is relatively low and gradually increases with an increase in the number of epochs. AAE and DCGAN outperform the rest of the variants, with AAE performing better in CS, PSNR, and MMD, whereas DCGAN outperforms the rest in FID and IS. InfoGAN and CGAN perform reasonably well, with CS of more than 0:83, followed by WGAN with a CS of 0:7270. Additionally, an example of memory GAN was tested for over-fitting. The *C*2 and*C*3 accuracy of a memory GAN are computed as 0:9232 and 0:9038, respectively. The RMS value of *C*2 and *C*3 is 0:9135, which is more than the validation accuracy of 0:9030, proving the occurrence of over-fitting.

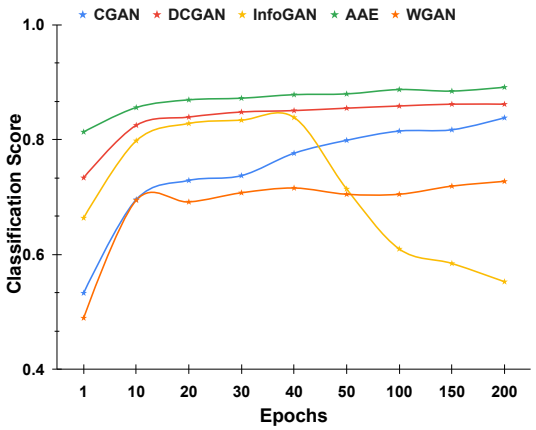

Fig. 10. Classification Score Vs Number of Epochs for F-MNIST

Similar to MNIST, each class in Fashion-MNIST has 6000 image samples. A different distribution is considered for the computation of DS to see whether GAN learns the contrasting distributions. 5000 sample images are chosen randomly from each of the first five classes. Similarly, 3000 sample images are randomly chosen from each of the last five classes to form new training data set S_t^0 of 40000 images. It can be seen in *fig:*12 that each variant of GAN learns the distribution quite well from an early stage in GAN training. AAE outperforms the other variants, with DCGAN and CGAN following closely.

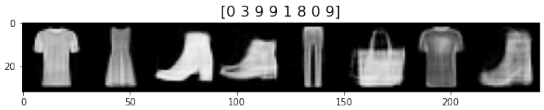

Fig. 11. FMNIST images generated from AAE and subsequent labelling done by *C*1

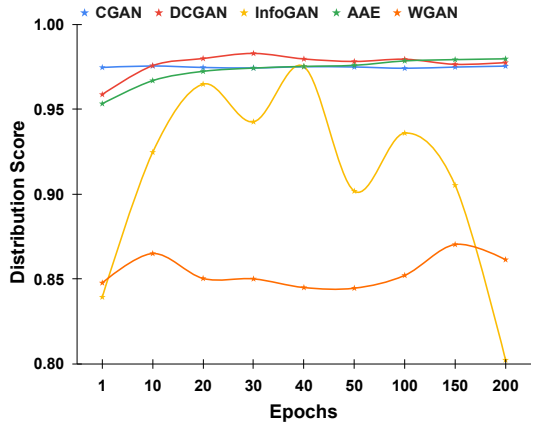

Fig. 12. Distribution Score Vs Number of Epochs for F-MNIST

3) CIFAR-10: A seven-layer CNN is trained on CIFAR-10 images for 25 epochs with a learning rate of 0:001 and used as the pre-trained model for image classification. The validation accuracy for CIFAR-10 is found to be 0:8449 and the classified images along with their respective classes are shown in *fig:*14. All the variants of GAN are trained for 1000 epochs, and the scores at the point of convergence are considered for comparative performance analysis. CS, DS FID, IS, PSNR, and MMD are calculated. It can be observed from *Table* IV and *fig:*13 that, similar to the other two datasets, CS for each variant is relatively low and gradually increases with an increase in the number of epochs. DCGAN outperforms the rest of the variants with the best CS, FID and IS score, with AAE closely following it in CS while outperforming other variants in PSNR and MMD. Additionally, an example of memory GAN was tested for over-fitting in for CIFAR-10 dataset. The *C*2 and*C*3 accuracy of the memory GAN are computed as 0:876 and 0:8543, respectively. The RMS value of *C*2 and *C*3 is 0:8652, which is more than the validation accuracy of 0:8449, proving the occurrence of over-fitting.

Each class in the training set of the CIFAR-10 data set has 5000 image samples. A different distribution is considered for

TABLE III
PERFORMANCE EVALUATION OF VARIOUS GANS ON FASHION-MNIST DATA SET

Fashion-MNIST (Validation accuracy from C1=0.9030)										
Metrics ! / **GANs #**	DC	IC	BC	**CS**	**DS**	**FID**	**IS**	**PSNR**	**MMD**	**Mode collapse or Over-fitting**
CGAN	0.8446	0.8147	0.8535	0.8377	0.9755	210.67	2.20	56.37	0.1256	None
DCGAN	0.8483	**0.8871**	0.8483	0.8614	0.9765	**122.63**	**2.78**	56.30	0.1240	None
InfoGAN	0.8200	0.8737	0.8201	0.8383	0.9752	142.99	2.39	55.98	0.1260	None
AAE	**0.8925**	0.8795	**0.9008**	**0.8910**	**0.9798**	125.49	1.99	**69.80**	**0.1106**	None
WGAN	0.6674	0.8430	0.6553	0.7270	0.8614	276.91	2.14	56.08	0.1281	None

TABLE IV
PERFORMANCE EVALUATION OF VARIOUS GANS ON CIFAR-10 DATA SET

CIFAR-10 (Validation accuracy from C1=0.8449)										
Metrics ! / **GANs #**	DC	IC	BC	**CS**	**DS**	**FID**	**IS**	**PSNR**	**MMD**	**Mode collapse or Over-fitting**
CGAN	0.4004	0.3948	**0.4591**	0.4191	**0.9748**	309.76	1.39	58.17	0.1248	None
DCGAN	0.5061	**0.4660**	0.3378	**0.4426**	0.9550	**218.87**	**1.73**	58.15	0.1249	None
InfoGAN	0.4629	0.4176	0.3302	0.4074	0.9156	234.83	1.66	57.39	0.1255	None
AAE	**0.5474**	0.3052	0.3376	0.4110	0.8333	238.44	1.66	**70.03**	**0.1206**	None
WGAN	0.4267	0.2854	0.3047	0.3447	0.8731	370.12	1.48	57.50	0.1259	None

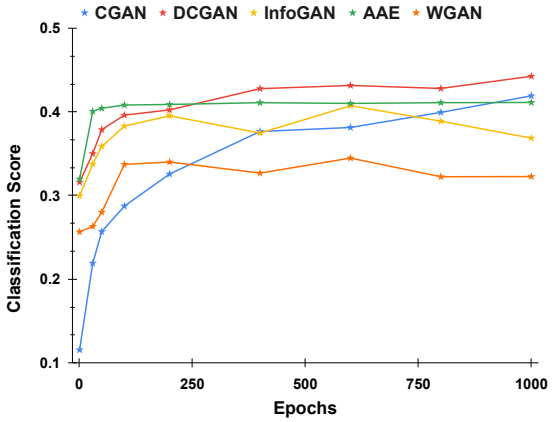

Fig. 13. Classification Score Vs Number of Epochs for CIFAR-10

Fig. 14. CIFAR-10 images generated from DCGAN and subsequent labelling done by C1

the computation of DS to see whether GAN learns the contrasting distributions. All 5000 sample images are considered from each of the first five classes, and 3000 sample images are randomly chosen from each of the last five classes to form new training data set S_t^0 of 40000 images. It can be seen in *fig:*15 that each variant of GAN learns the distribution quite well from an early stage in GAN training. CGAN outperforms the other variants, with DCGAN and InfoGAN following closely.

C. Discussion

The CS of all the variants except AAE is observed to be comparatively lower for initial epochs. This is because,

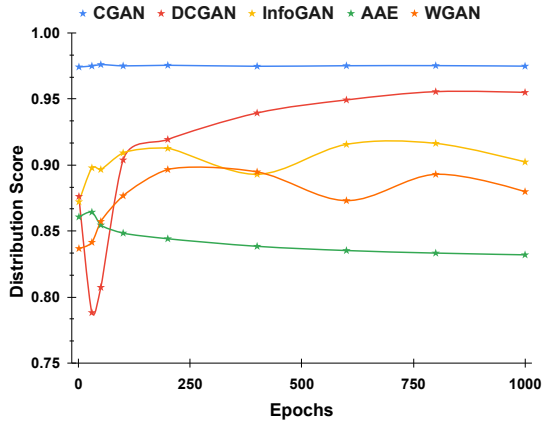

Fig. 15. Distribution Score Vs Number of Epochs for CIFAR-10

initially, the generator has no access to the real images and tries to produce images from latent noise. As the number of epochs increases, GANs perform better as the generator learns through fine-tuning. It can also be observed that AAE outperforms the rest of the variants in both data sets. The major reason behind the better performance of AAE is the autoencoder acting as the generator, which can learn and produce realistic fake images. Unlike other variants of GAN, in AAE, the autoencoder has access to real images. The decoder actually reconstructs the images from encrypted data rather than generating fake images from latent noise. Similarly, CGAN also performs well in MNIST because of the conditional labels given to the generator. Initially, the generator in CGAN fails to produce impressive images, but it learns quickly as images are associated with labels that help produce better images in subsequent epochs.

It can also be observed that once the GANs converge, the learning of the generator becomes negligible, and the

CS becomes stagnant and does not fluctuate much except in the case of InfoGAN. In InfoGAN, only salient features of the images are extracted and learned, and images are produced based on these primary features. So, after the GANs converged, subsequent epochs resulted in information loss, and the quality and diversity of the generated images worsened further, resulting in a dip in the performance.

Three examples, one each of memory GAN for MNIST, F-MNIST, and CIFAR-10, are considered to show over-fitting detection. It can be seen that over-fitting can be successfully detected in all the cases. Both DC and IC values are unusually higher in memory GANs as it simply memorizes the real images and does not produce any new different images. The RMS value of DC and IC is more than the validation accuracy, which is the upper threshold for over-fitting in GANs.

The reliance on pre-trained CNNs for GAN performance evaluation may introduce a dependency on classifier quality. The effectiveness of the metrics can be limited by the performance of the classifier employed. Addressing this dependency becomes important, as inaccuracies in the classifier can result in erroneous metric assessments. To address this challenge, continuous evaluation of the classifier model, along with regular updates and ensemble techniques, can bolster the robustness of the metrics. Furthermore, fine-tuning the classifier on task-specific data and leveraging transfer learning techniques can enhance classifier accuracy, thereby reducing reliance on initial pre-training. By proactively addressing these concerns, the metrics can yield more dependable assessments, despite their reliance on classifier models.

VII. Conclusion

In this work, we presented two CNN-based measures, CS and DS, for the performance evaluation of GAN variants. A novel method to detect over-fitting in GANs through image classification was presented. A novel method to detect mode collapse through image classification was also proposed. Comparative performance analysis of 5 different variants of GAN as CGAN, DCGAN, InfoGAN, AAE, and WGAN was done for MNIST, F-MNIST and CIFAR-10 data sets. Previous findings that AAE outperforms the rest of the GAN variants in generating synthetic images of the MNIST and F-MNIST data sets were corroborated through CS and DS. CS and DS were also compared with the most widely used GAN evaluation techniques, and it was shown that both metrics could evaluate the performance of GANs quantitatively. The qualitative performance evaluation by our proposed methods was also shown by successfully detecting over-fitting in memory GAN for all three data sets. Overall, it was found that CS and DS overcame the inherent drawbacks of existing methods and could evaluate different variants of GANs quantitatively and qualitatively.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680 (2014).

[2] M. Pieters, M. Wiering, Comparing generative adversarial network techniques for image creation and modification, arXiv preprint arXiv:1803.09093 (2018).

[3] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A. A. Bharath, Generative adversarial networks: An overview, IEEE Signal Processing Magazine 35 (1) (2018) 53–65 (2018).

[4] A. Borji, Pros and cons of gan evaluation measures, Computer Vision and Image Understanding 179 (2019) 41–65 (2019).

[5] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, D. Jurafsky, Adversarial learning for neural dialogue generation, arXiv preprint arXiv:1701.06547 (2017).

[6] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, R. Webb, Learning from simulated and unsupervised images through adversarial training, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2107–2116 (2017).

[7] H. Huang, P. S. Yu, C. Wang, An introduction to image synthesis with generative adversarial nets, arXiv preprint arXiv:1803.04469 (2018).

[8] X. Wang, A. Shrivastava, A. Gupta, A-fast-rcnn: Hard positive generation via adversary for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2606–2615 (2017).

[9] L. Yu, W. Zhang, J. Wang, Y. Yu, Seqgan: Sequence generative adversarial nets with policy gradient, in: Thirty-first AAAI conference on artificial intelligence, 2017 (2017).

[10] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125–1134 (2017).

[11] T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, M. Sun, Show, adapt and tell: Adversarial training of cross-domain image captioner, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 521–530 (2017).

[12] X. Liang, Z. Hu, H. Zhang, C. Gan, E. P. Xing, Recurrent topic-transition gan for visual paragraph generation, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 3362–3371 (2017).

[13] W. Zhao, W. Xu, M. Yang, J. Ye, Z. Zhao, Y. Feng, Y. Qiao, Dual learning for cross-domain image captioning, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 29–38 (2017).

[14] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, M. N. Do, Semantic image inpainting with deep generative models, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5485–5493 (2017).

[15] H. Huang, P. S. Yu, C. Wang, An introduction to image synthesis with generative adversarial nets, arXiv preprint arXiv:1803.04469 (2018).

[16] P. Samangouei, M. Kabkab, R. Chellappa, Defense-gan: Protecting classifiers against adversarial attacks using generative models, arXiv preprint arXiv:1805.06605 (2018).

[17] L. Theis, A. v. d. Oord, M. Bethge, A note on the evaluation of generative models, arXiv preprint arXiv:1511.01844 (2015).

[18] J. Snell, K. Ridgeway, R. Liao, B. D. Roads, M. C. Mozer, R. S. Zemel, Learning to generate images with perceptual similarity metrics, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE, 2017, pp. 4277–4281 (2017).

[19] A. Oliva, Gist of the scene, in: Neurobiology of attention, Elsevier, 2005, pp. 251–256 (2005).

[20] Y. Wu, Y. Burda, R. Salakhutdinov, R. Grosse, On the quantitative analysis of decoder-based generative models, arXiv preprint arXiv:1611.04273 (2016).

[21] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Advances in neural information processing systems, 2016, pp. 2234–2242 (2016).

[22] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, in: Advances in neural information processing systems, 2017, pp. 6626–6637 (2017).

[23] H. Alqahtani, M. Kavakli-Thorne, G. Kumar, F. SBSSTC, An analysis of evaluation metrics of gans, in: International Conference on Information Technology and Applications (ICITA), 2019 (2019).

[24] R. Fortet, E. Mourier, Convergence de la répartition empirique vers la répartition théorique, in: Annales scientifiques de l'École Normale Supérieure, Vol. 70, 1953, pp. 267–285 (1953).

[25] D. J. Im, C. D. Kim, H. Jiang, R. Memisevic, Generating images with recurrent adversarial networks, arXiv preprint arXiv:1602.05110 (2016).

[26] M. Lucic, K. Kurach, M. Michalski, S. Gelly, O. Bousquet, Are gans created equal? a large-scale study, in: Advances in neural information processing systems, 2018, pp. 700–709 (2018).