axioms

MDPI

Article

Score-Guided Generative Adversarial Networks

Minhyeok Lee [1] and Junhee Seok [2,*]

[1] School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 06974, Republic of Korea
[2] School of Electrical Engineering, Korea University, Seoul 02841, Republic of Korea
* Correspondence: jseok14@korea.ac.kr

Abstract: We propose a generative adversarial network (GAN) that introduces an evaluator module using pretrained networks. The proposed model, called a score-guided GAN (ScoreGAN), is trained using an evaluation metric for GANs, i.e., the Inception score, as a rough guide for the training of the generator. Using another pretrained network instead of the Inception network, ScoreGAN circumvents overfitting of the Inception network such that the generated samples do not correspond to adversarial examples of the Inception network. In addition, evaluation metrics are employed only in an auxiliary role to prevent overfitting. When evaluated using the CIFAR-10 dataset, ScoreGAN achieved an Inception score of 10.36 ± 0.15, which corresponds to state-of-the-art performance. To generalize the effectiveness of ScoreGAN, the model was evaluated further using another dataset, CIFAR-100. ScoreGAN outperformed other existing methods, achieving a Fréchet Inception distance (FID) of 13.98.

Keywords: generative adversarial network; image generation; image synthesis; GAN; generative model; Inception score; scoreGAN

MSC: 68T45

check for updates

Citation: Lee, M.; Seok, J. Score-Guided Generative Adversarial Networks. *Axioms* **2022**, *11*, 701. https://doi.org/10.3390/axioms11120701

Academic Editor: Joao Paulo Carvalho

Received: 3 November 2022
Accepted: 3 December 2022
Published: 7 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A recent advancement in artificial intelligence is the implementation of deep learning algorithms to generate synthetic samples [1–3]. These types of neural networks are able to learn how to map inputs to outputs after being trained on large datasets. In the past few years, researchers have used deep learning algorithms to create synthetic samples in various domains such as music, images, and speech [4–6]. One important application of synthetic sample generation is in the field of data augmentation [3,7]. Data augmentation is a technique used in machine learning to increase the size of the training datasets. Synthetic samples can be used to create new data points that are similar to existing data points, but may have different labels or attributes. This can help improve the performance of machine learning algorithms by providing them with more data to train on.

Due to their innovative training algorithm and superb performance in image generation tasks, generative adversarial networks (GANs) have been widely studied in recent years [8–12]. GANs generally employ two artificial neural network (ANN) modules, called a generator and a discriminator, which are trained with an adversarial process to detect and deceive each other. Specifically, the discriminator aims at detecting synthetic samples that are produced by the generator; meanwhile, the generator is trained by errors that are obtained from the discriminator. By such a competitive learning process, the generator can produce fine synthetic samples of which features are incredibly similar to those of actual samples [13,14].

However, the performance evaluation of GAN models is a challenging task since the quality and diversity of generated samples should be assessed from the human perspective [15,16]; furthermore, unbiased evaluations are also difficult because each person can have different

views on the quality and diversity of samples. Therefore, several studies have introduced quantitative metrics to evaluate GAN models in a measurable manner [16,17].

The Inception score is one of the most representative metrics to evaluate GAN models for image generation [16]. A conventional pretrained ANN model for image classification, called the Inception network [18], is employed to assess both the quality and diversity of the generated samples, by measuring entropies of inter- and intra-samples in terms of estimated probabilities for each class. The Fréchet Inception distance (FID) is another metric to measure GAN performance, in which the distance between feature distributions of real samples and generated samples are calculated [17].

From the adoption of the evaluation metrics, the following questions then arise: Can the evaluation metrics be used as targets for the training of GAN modelssince the metrics reasonably represent the quality and diversity of samples? By backpropagating gradients of the score or distance, is it possible to maximize or minimize them? Such an approach seems feasible since the metrics are generally differentiable; therefore, the gradients can be computed and backpropagated.

However, simply backpropagating the gradients and training with the metrics correspond to learning adversarial examples in general [19,20]. Since the complexity of ANN models is significantly high, we can easily make a sample be incorrectly predicted, by adding minimal noises into the sample; this noisy sample is called the adversarial example [20]. Therefore, in short, a fine quality and rich diversity of samples can have a high Inception score, while the reverse is not always true.

Barratt and Sharma [21] studied this problem and found that directly maximizing the score does not guarantee that the generator produces fine samples. They trained a GAN model to maximize the Inception score; then, the trained model produced image samples with a very high Inception score. While the Inception score of real samples in the CIFAR-10 dataset is around 10.0, the produced images achieved an Inception score of 900.15 [21]. However, the produced images were entirely different from the real images in the CIFAR-10 dataset; instead, they looked like noises.

In this paper, to address such a problem and utilize the evaluation metric as a training method, we propose a score-guided GAN (ScoreGAN) that employs an evaluator ANN module using pretrained networks with the evaluation metrics. While the aforementioned problems exist in ordinary GANs, ScoreGAN solves the problems through two approaches as follows.

First, ScoreGAN uses the evaluation metric as an auxiliary target, while the target function of ordinary GANs is mainly used. Using the evaluation metric as the only target causes overfitting of the network used for the metric, instead of learning meaningful information from the network, as shown in related studies [21]. Thus, the evaluation metric is employed as the auxiliary target in ScoreGAN.

Second, in order to backpropagate gradients and train the generator in ScoreGAN, we employ a different pretrained model called MobileNet [22]. This prevents the generator from overfitting on the Inception network. To the best of our knowledge, employing a pretrained MobileNet with an additional score function for the training of the generator has not been explored thus far. Additionally, this approach allows us to validate that the generator has actually learned features, rather than simply memorizing details from the Inception network. In this process, we can assess whether ScoreGAN is able to achieve a high Inception score without using the Inception network, which can prove the effectiveness of ScoreGAN.

The main contributions of this paper are as follows:

- The score-guided GAN (ScoreGAN) that uses the evaluation metric as an additional target is proposed.
- The proposed ScoreGAN circumvents the overfitting problem by using MobileNet as an evaluator.
- Evaluated by the Inception score and cross-validated through the FID, ScoreGAN demonstrates state-of-the-art performance on the CIFAR-10 dataset and CIFAR-100

dataset, where its Inception score in the CIFAR-10 is 10.36 ± 0.15, and the FID in the CIFAR-100 is 13.98.

2. Background

Generative models aim to learn sample distributions and produce realistic samples. For instance, generative models can be trained with an image dataset; then, a successfully trained generative model produces realistic, but synthetic images for which the features are extremely similar to the original images in the training set. The GAN is one the representative generative models, which uses deep learning architectures and an algorithm with game theory. In recent years, diffusion models have been employed as generative models and demonstrated superior performances [2,23,24]. In Section 2.1, we discuss a variant of the GAN called the controllable GAN, which is the baseline of the proposed model. Additionally, two metrics to assess the produced images by the generative models are presented in Sections 2.2 and 2.3.

2.1. Controllable Generative Adversarial Networks

The conventional GAN model consists of two ANN modules, i.e., the generator and the discriminator. The two modules are trained by playing a game to deceive or detect each other [15,25]. The game to train a GAN can be represented as follows:

$$\hat{\boldsymbol{\theta}}_D = \arg\min_{\boldsymbol{\theta}_D} \{ L_D(1, D(X; \boldsymbol{\theta}_D)) + L_D\big(0, D\big(G\big(Z; \hat{\boldsymbol{\theta}}_G\big); \boldsymbol{\theta}_D\big)\big) \}, \tag{1}$$

$$\hat{\boldsymbol{\theta}}_G = \arg\min_{\boldsymbol{\theta}_G} \{ L_D(1, D(G(Z; \boldsymbol{\theta}_G); \boldsymbol{\theta}_D)) \}, \tag{2}$$

where G and D denote the generator and the discriminator, respectively, X is a training sample, Z represents a noise vector, $\boldsymbol{\theta}$ is a set of weights of an ANN model, and L_D indicates a loss function for the discriminator.

However, the ordinary GAN can hardly produce the desired samples since each feature in a dataset is randomly mapped into each variable of the input noise vector. Therefore, it is hard to discover which noise variable corresponds to which feature. To overcome this problem, conditional variants of GAN that introduce conditional input variables have been studied [26–28].

Controllable GAN (ControlGAN) [29] is one of the conditional variants of GANs that uses an independent classifier and the data augmentation techniques to train the classifier. While a conventional model, called auxiliary classifier GAN (ACGAN) [28], has an overfitting issue on the classification loss and a trade-off for using the data augmentation technique [29], ControlGAN breaks the trade-off through introducing the independent classifier, as well as the data augmentation technique. The training of ControlGAN is performed as follows:

$$\hat{\boldsymbol{\theta}}_D = \arg\min_{\boldsymbol{\theta}_D} \{ L_D(1, D(X; \boldsymbol{\theta}_D)) + L_D\big(0, D\big(G\big(Z, \mathcal{L}; \hat{\boldsymbol{\theta}}_G\big); \boldsymbol{\theta}_D\big)\big) \}, \tag{3}$$

$$\hat{\boldsymbol{\theta}}_G = \arg\min_{\boldsymbol{\theta}_G} \{ L_D(1, D(G(Z; \boldsymbol{\theta}_G); \boldsymbol{\theta}_D)) + \gamma_t \cdot L_C\big(\mathcal{L}, C\big(G(Z, \mathcal{L}; \boldsymbol{\theta}_G); \hat{\boldsymbol{\theta}}_C\big)\big) \}, \tag{4}$$

$$\hat{\boldsymbol{\theta}}_C = \arg\min_{\boldsymbol{\theta}_C} \{ L_C(\mathcal{L}, C(X; \boldsymbol{\theta}_C)) \}, \tag{5}$$

where C represents the independent classifier, $\mathcal{L}$ denotes the input labels, and γ_t is a learning parameter that modulates the training of the generator in terms of the classification loss.

2.2. The Inception Score

To assess the quality and diversity of the generated samples by GANs, the Inception score [16] is one of the most conventional evaluation metrics, which has been extensively employed in many studies [8,14,16,21,26,27,29]. For the quantitative evaluation of GANs,

the Inception score introduces the Inception network, which was initially used for image classification [18]. The Inception network is pretrained to solve the image classification task over the ImageNet dataset [30], which contains more than one million images of 1000 different classes; then, the network learns the general features of various objects.

Through the pretrained Inception network, the quality and diversity of the generated samples can be obtained from two aspects [16,21]: First, the high quality of an image can be guaranteed if the image is firmly classified into a specific class. Second, a high entropy in the marginal probability of the generated samples indicates a rich diversity of the samples since such a condition signifies that the generated samples are different from each other.

Therefore, the entropies of the intra- and inter-samples are calculated over the generated samples; then, these two entropies compose the Inception score as follows:

$$IS\left(G\left(\cdot;\hat{\boldsymbol{\theta}}_G\right)\right) = \exp\left(\frac{1}{N}\sum KL\left(Pr\left(Y|\hat{X}\right)||Pr(Y)\right)\right), \tag{6}$$

where $\hat{X}$ denotes a generated sample, KL indicates the Kullback–Leibler (KL) divergence, namely the relative entropy, and N is the number of samples in a batch. Since a high KL divergence signifies a significant difference between the two probabilities, thus a higher Inception score indicates greater qualities and a wider variety of samples. Generally, ten sets, each of which contains 5000 generated samples, are used to calculate the Inception score [16,21].

2.3. The Fréchet Inception Distance

The FID is another metric to evaluate the generated samples in which the Inception network is employed as well [17]. Instead of the predicted probabilities, the FID introduces the feature distribution of the generated samples that can be represented as the outputs of the penultimate layer of the Inception network.

With the assumption that the feature distribution follows a multivariate normal distribution, the distance between the feature distributions of the real samples and generated samples is calculated as follows:

$$FID\left(\boldsymbol{X},\hat{\boldsymbol{X}}\right) = \left\|\mu_{\boldsymbol{X}} - \mu_{\hat{\boldsymbol{X}}}\right\|_2^2 + Tr\left(\Sigma_{\boldsymbol{X}} + \Sigma_{\hat{\boldsymbol{X}}} - 2\cdot\sqrt{\Sigma_{\boldsymbol{X}}\Sigma_{\hat{\boldsymbol{X}}}}\right), \tag{7}$$

where $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ are the data matrices of the real samples and generated samples, respectively, and Σ denotes the covariance matrix of a data matrix. In contrast to the Inception score, a lower FID indicates the similarity between the feature distributions since the FID measures a distance.

3. Methods

In this paper, we propose ScoreGAN, which uses an additional target, derived from the evaluation metrics in Section 2.2. The proposed ScoreGAN uses the Inception score as a target of the generator. However, directly targeting the Inception score leads to an overfitting issue; thus, in ScoreGAN, a pretrained MobileNet is used for the training. Then, the trained model is evaluated with the conventional Inception score and FID using the Inception network. This method is elaborated in Section 3.1. The training details of ScoreGAN are described in Section 3.2.

3.1. Score-Guided Generative Adversarial Network

The main idea of ScoreGAN is straightforward: For its training, the generator in ScoreGAN utilizes an additional loss that can be obtained from the evaluation metric for GANs. Since it has been verified that the evaluation metric strongly reflects the quality and diversity of the generated samples [8,16], it is expected that the performance of GAN models can be enhanced by optimizing the metrics.

Therefore, the architecture of ScoreGAN corresponds to ControlGAN with an additional evaluator; the evaluator is used to calculate the score, then gradients are backpropagated to train the generator. The other neural network structures are the same as those of ControlGAN.

However, due to the high complexity of GANs, it is not guaranteed that such an approach can work properly, as described in the previous section. Directly optimizing the Inception score can cause overfitting over the network that is used to compute the metric; then, the overfitted GANs produce noises instead of realistic samples even if the score of the generated noise is high [21].

In this paper, we circumvent this problem through two different approaches, i.e., employing the metric as an auxiliary cost instead of the main target of the generator and adopting another pretrained network as an evaluator module as a replacement of the Inception network.

3.1.1. The Auxiliary Costs Using the Evaluation Metrics

ScoreGAN mainly uses the ordinary GAN cost in which the adversarial training process is performed while the evaluation metric is utilized as an auxiliary cost. Therefore, the training of the generator in ScoreGAN is conducted by adding the cost of the evaluation metric to (4). Such a method using an auxiliary cost has been introduced in ACGAN [28]; then, the method has been widely studied in many recent works [27], including ControlGAN [29]. As a result of the recent works, it has been demonstrated that the auxiliary costs serve as a "rough guide" for a generator to be trained with additional information. The proposed technique using the evaluation metrics in this paper corresponds to a variant of such a method, where the metrics are used as rough guides to generate high-quality and a rich variety of samples. In short, the generator in ScoreGAN aims at maximizing a score in addition to the original cost, which can be represented as follows:

$$\hat{\boldsymbol{\theta}}_G = \arg\min_{\boldsymbol{\theta}_G} \{ \mathfrak{L}_G - \delta \cdot IS(\hat{\boldsymbol{X}}) \}, \tag{8}$$

where $\mathfrak{L}_G$ denotes the regular cost for a generator, such as the optimization target in (4), δ is a parameter for the score, and IS is the score that can be obtained from the evaluator. Since (6) is differentiable with respect to G, $\boldsymbol{\theta}_G$ can be optimized by the gradients in such a manner.

3.1.2. The Evaluator Module with MobileNet

To obtain the IS in (8), originally, the Inception network [18] is required as the evaluator in ScoreGAN since the metrics are calculated through the network. However, as described in the previous sections, directly optimizing the score leads to overfitting the network, thereby making the generator produce noises instead of fine samples. Furthermore, if the Inception network is used for the training, it is challenging to validate whether the generator actually learns features rather than memorizes the network, since the generator trained by the Inception network certainly achieves a high Inception score, regardless of the actual learning.

Therefore, ScoreGAN introduces another network, called MobileNet [22], as the evaluator module, in order to maximize the score. MobileNet [22,31,32] is a comparatively small classifier for mobile devices, which is trained with the ImageNet dataset as well. Due to its compact network size, enabling GANs to be trained, MobileNet is used in this study. The score is calculated over the feature distribution of MobileNet; then, the generator aims to maximize the score, as described in (8). For MobileNet, the pretrained model in the Keras library is used in this study.

Furthermore, to prevent overfitting on MobileNet, ScoreGAN uses a regularized score, which can be represented as follows:

$$RIS_{mobile}(\hat{\boldsymbol{X}}) := \min\{ IS_{mobile}(\boldsymbol{X}), IS_{mobile}(\hat{\boldsymbol{X}}) \}, \tag{9}$$

where RIS represents the regularized score and IS_{mobile} denotes the score calculated by the same manner as (6) through MobileNet instead of the Inception network. Since a perfect GAN model can achieve a high score that is similar to the score of real data, thus, it is expected that the maximum value of the score that a GAN model can attain is the score of real data. Therefore, such an approach in (9) assists the GAN training by reducing the overfitting of the target network.

The evaluation, however, is performed with the Inception network, as well as the Inception score, instead of MobileNet and IS_{mobile}, which can generalize the performance of ScoreGAN. If ScoreGAN is trained to optimize MobileNet, the training ensures maximizing the score obtained with MobileNet, irrespective of the learning of actual features. Therefore, to validate the performance, the model must be evaluated with the original metric, the Inception score.

Furthermore, the model is further evaluated and cross-validated through the FID. Since the score and the FID measure different aspects of the generated samples, the maximization of the score does not guarantee obtaining a low FID. Instead, only if ScoreGAN produces realistic samples that are highly similar to real data in terms of feature distributions, the model can achieve a lower FID than the baseline. Therefore, by using the FID, we can properly cross-validate the model even if the score is used for the target.

3.2. Network Structures and Regularization

Since ScoreGAN employs the ControlGAN structure as the baseline and integrates an evaluator measuring the score with the baseline, ScoreGAN consists of four ANN modules, namely the generator, discriminator, classifier, and evaluator. In short, ScoreGAN additionally uses the evaluator, attached to the original ControlGAN framework. The structure of ScoreGAN is illustrated in Figure 1.

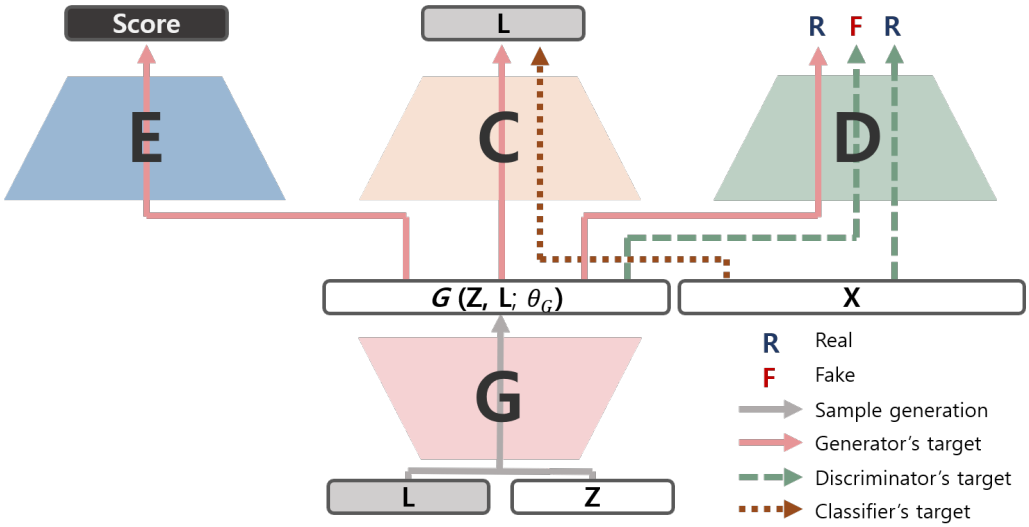

Figure 1. The structure of ScoreGAN.The training of each module is represented with arrows. E: evaluator; C: classifier; D: discriminator; G: generator.

As described in Figure 1 and (8), the generator is trained by targeting the three other ANN modules to maximize the score and minimize the losses, simultaneously. Meanwhile, the discriminator tries to distinguish between the real samples and generated samples. The classifier is trained only with the real samples in which the data augmentation is applied; then, the loss for the generator can be obtained with the trained classifier. The evaluator is a pretrained network and fixed during the training of the generator; thereby, the generator learns general features of various objects from the pretrained evaluator by maximizing the score of the evaluator.

Due to the vulnerable nature of the training of GANs, regularization methods for the ANN modules in GANs are essential [33,34]. Accordingly, ScoreGAN also uses the regularization methods that are widely employed in various GAN models for its training. Spectral normalization [35] and the hinge loss [36] that are commonly used in state-of-the-art GAN models are employed in ScoreGAN as well. The gradient penalty with a weight parameter of 10 is used [33]. Furthermore, according to recent studies that show the regularized discriminator requires intense training [8,35], multiple training iterations for the discriminator are applied; the discriminator is trained over five times per one training iteration of the generator. For the generator and the classifier, the conditional batch normalization (cBN) [37] and layer normalization (LN) [38] techniques are used, respectively.

For the neural network structures in ScoreGAN, we followed a typical architecture that is generally introduced in many other studies [27,39]. The detailed structures are shown in Table 1. Two time-scale update rule (TTUR) [17] is employed with learning rates of 4×10^{-4} and 2×10^{-4} for the discriminator and the generator, respectively. The learning rates halve after 50,000 iterations; then, the models are further trained with the halved learning rates for another 50,000 iterations. The Adam optimization method is used with the parameters of $\beta_1 = 0$ and $\beta_2 = 0.9$, which is the same setting as the other recent studies [29,35]. The maximum threshold for the training from the classifier was set to 0.1. The parameter δ in (8) that modulates the training from the evaluator was set to 0.5.

Table 1. Architecture of neural network modules. The values in the brackets indicate the number of convolutional filters or nodes of the layers. Each ResBlock is composed of two convolutional layers with pre-activation functions.

Generator	Discriminator	Classifier
$Z \in \mathbb{R}^{128}$	$Z \in \mathbb{R}^{32\times32\times3}$	$Z \in \mathbb{R}^{32\times32\times3}$
Dense $(4 \times 4 \times 256)$	ResBlock Downsample (256)	ResBlock (32) $\times$ 3 ResBlock Downsample (32)
ResBlock Upsample (256)	ResBlock Downsample (256)	ResBlock (64) $\times$ 3 ResBlock Downsample (64)
ResBlock Upsample (256)	ResBlock (256)	ResBlock (128) $\times$ 3 ResBlock Downsample (128)
ResBlock Upsample (256)	ResBlock (256)	ResBlock (128) $\times$ 3
cBN; ReLU; Conv (3); Tanh	ReLU; Global Pool; Dense (1)	LN; ReLU; Global Pool; Dense (10)

4. Results

In this section, we discuss the performance of ScoreGAN with respect to the Inception score, the FID, and the quality of the generated images. In the experiments, three images datasets called CIFAR-10, CIFAR-100, and LSUN were used. Three subsections in this section explain the performance results on each dataset. The characteristics of the datasets are described in Table 2.

Table 2. Datasets used in the experiments.

Name	Image Res.	No. of Samples	Descriptions
CIFAR-10	32×32	50,000	10 classes of small objects 5000 images per class
CIFAR-100	32×32	50,000	100 classes of small objects 500 images per class
LSUN	down-sampled to 128×128	around 10 million	10 classes of indoor and outdoor scenes around 120,000 to 3,000,000 per class

4.1. Image Generation with CIFAR-10 Dataset

The proposed ScoreGAN was evaluated over the CIFAR-10 dataset, which is conventionally employed as a standard dataset to assess the image generation performance of GAN models in many studies [26,27,29,35,39–42]. The training set of the CIFAR-10 dataset is composed of 50,000 images that are from 10 different classes. To train the models, we used a minibatch size of 64, and the generator was trained over 100,000 iterations. The other settings and the structure of ScoreGAN that was used to train the CIFAR-10 dataset are described in the previous section. Since the proposed ScoreGAN introduces an additional evaluator compared to ControlGAN, we used ControlGAN as the baseline; thereby, we can properly assess the effect of the additional evaluator.

To evaluate the image generation performance of the models, the Inception score and FID were employed. As described in the previous sections, since the Inception score is the average of the relative entropy between each prediction and the marginal predictions, a higher Inception score signifies better-quality and a rich diversity of the generated samples; conversely, a lower FID indicates that the feature distributions of the generated samples are similar to those of the real samples. Notice that, for ScoreGAN, the Inception score and FID are measured after the training iterations (100,000). It is expected that we can enhance the performance results if the models are repeatably measured during the training, and then, we selected the best model among the iterations, as conducted in several studies [8,39].

Table 3 shows the performance of GAN models in terms of the Inception score and FID. While the neural network architectures of the GAN are the same as ControlGAN, the proposed ScoreGAN demonstrates superior performance compared to ControlGAN, which verifies the effectiveness of the additional evaluator in ScoreGAN. The Inception score increased by 20.5%, from 8.60 to 10.36, which corresponds to state-of-the-art performance among the existing models thus far. The FID also decreased by 21.1% in ScoreGAN compared to ControlGAN in which the FID values of ScoreGAN and ControlGAN are 8.66 and 10.97, respectively. Random examples that are generated by ScoreGAN are shown in Figure 2.

Figure 2. Random examples of the generated images by ScoreGAN with the CIFAR-10 dataset. Each column represents each class in the CIFAR-10 dataset. All images have a 32 × 32 resolution.

Table 3. Performance of GAN models over the CIFAR-10 dataset. IS indicates the Inception score; FID indicates the Fréchet Inception distance. The best performances are highlighted in bold.

Methods	IS	FID
Real data	11.23 ± 0.20	-
ControlGAN [29]	8.61 ± 0.10	-
ControlGAN (w/Table 1; baseline)	8.60 ± 0.09	10.97
Conditional DCGAN [40]	6.58	-
AC-WGAN-GP [33]	8.42 ± 0.10	-
CAGAN [27]	8.61 ± 0.12	-
Splitting GAN [41]	8.87 ± 0.09	-
BigGAN [8]	9.22	14.73
MHingeGAN [39]	9.58 ± 0.09	**7.50**
ScoreGAN	**10.36 ± 0.15**	8.66

The results of this study appear to validate the effectiveness of both the additional evaluator and auxiliary score present in ScoreGAN. It can be said that the generator in ScoreGAN appears to properly learn general features through the pretrained evaluator and is then enforced to produce a variety of samples by maximizing the score. This is reflected not only in an increase in the Inception scores, but also in a decrease in the FID scores. Since the FID measures the similarity between feature distributions, it is less related to the objective of ScoreGAN. Therefore, this enhancement of the decreased FIDs could be evidence that ScoreGAN does not overfit on the Inception scores, and the proposed evaluator enhances the performance. Furthermore, since ScoreGAN does not use the Inception network as the evaluator and the score, it is difficult to regard the generated samples by ScoreGAN as adversarial examples of the Inception network, as shown in the examples in Figure 2, where the images are far from noises.

The detailed Inception score and FID over iterations are shown in Figure 3. As shown in the figures, the training of ControlGAN becomes slow after 30,000 iterations, while the proposed ScoreGAN continues its training. For example, the Inception score of ControlGAN at 35,000 iterations is 8.48, which is 98.6% of the final Inception score, while, at the same time, the Inception score of ScoreGAN is 9.34, which corresponds to 90.2% of its final score. The FID demonstrates similar results to those of the Inception score. In ControlGAN, the FID decreases by 10.7% from 50,000 to 100,000 iterations; in contrast, it declines by 26.9% in ScoreGAN. Such a result implies that the generator in ScoreGAN can be further trained by the proposed evaluator, although the training of the discriminator is saturated.

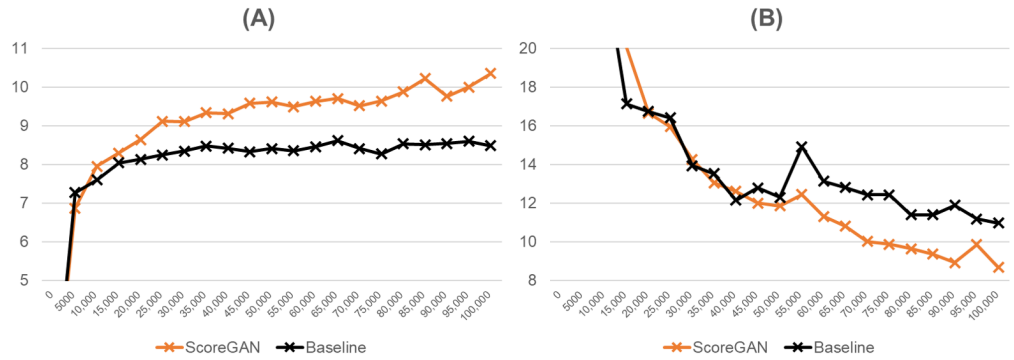

Figure 3. The performance of ScoreGAN in terms of the Inception score and Fréchet Inception distance over iterations. (**A**) The Inception scores; (**B**) the Fréchet Inception distance (FID). The baseline is ControlGAN with the same neural network architecture, identical to that of ScoreGAN.

4.2. Image Generation with CIFAR-100 Dataset

To generalize the effectiveness of ScoreGAN, the CIFAR-100 dataset was employed for the evaluation of the GAN models. The CIFAR-100 dataset is similar to the CIFAR-10 dataset, where each dataset contains 50,000 images of size 32×32 in the training set. The difference between the CIFAR-100 dataset and the CIFAR-10 dataset is that the CIFAR-100 dataset is composed of 100 different classes. Therefore, it is generally regarded that the training of the CIFAR-100 dataset is more challenging than that of the CIFAR-10 dataset. The architectures used in this experiment are shown in Appendix A.

Since existing methods in several recent studies have been evaluated over the CIFAR-100 dataset [43], we compared the performance between ScoreGAN and the existing methods. The performance in terms of the Inception score and FID is demonstrated in Table 4. The results show that ScoreGAN outperforms the other existing models. While the same neural network architectures are used in both methods, the performance of ScoreGAN is significantly superior to that of the baseline. For instance, the FID significantly declines from 18.42 to 13.98, which corresponds to a state-of-the-art result. Random examples of the generated images with ScoreGAN trained with CIFAR-100 are shown in Figure 4.

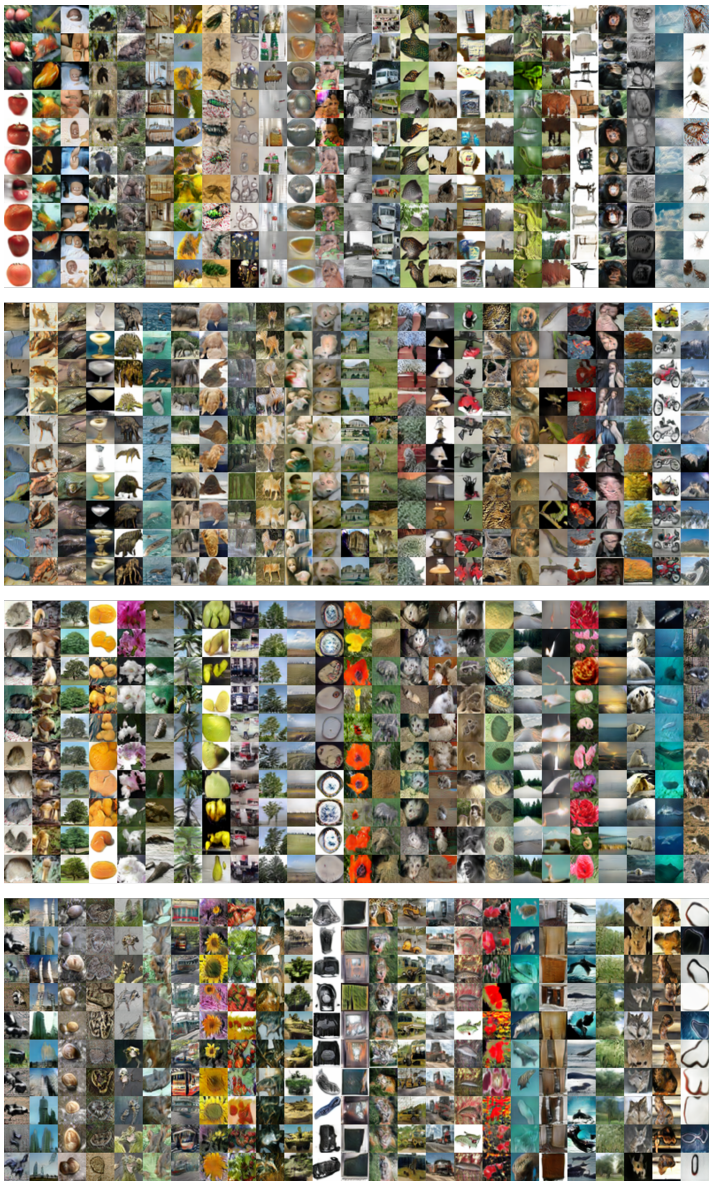

Figure 4. Random examples of the generated images by ScoreGAN with the CIFAR-100 dataset. Each column represents each class in the CIFAR-100 dataset. All images have a 32×32 resolution.

Table 4. Performance of the GAN models over the CIFAR-100 dataset. IS indicates the Inception score; FID indicates the Fréchet Inception distance. The best performances are highlighted in bold.

Methods	IS	FID
Real data	14.79 ± 0.18	-
ControlGAN (baseline)	9.32 ± 0.11	18.42
MSGAN [43]	-	19.74
SNGAN [42]	9.30 ± 0.08	15.6
MHingeGAN [39]	$\mathbf{14.36 \pm 0.09}$	17.30
ScoreGAN	13.11 ± 0.16	**13.98**

While the Inception score of ScoreGAN is slightly lower than that of MHingeGAN [39], such a disparity results from a difference in the assessment of the scores, in which, for MHingeGAN, the Inception score is continuously measured during the training iterations; then, the best score is selected among the training iterations. In contrast, the Inception score of ScoreGAN is computed only once after 100,000 iterations. Furthermore, in terms of the FID, ScoreGAN demonstrates superior results, compared to MHingeGAN. Furthermore, it is reported that the training of MHingeGAN over the CIFAR-100 dataset collapses before 100,000 iterations.

4.3. Image Generation with LSUN Dataset

For an additional experiment, ScoreGAN was applied to another dataset, called LSUN [44]. LSUN is a large-scale image dataset with 10 million images in 10 different scene categories, such as bedroom and kitchen. Furthermore, different from the CIFAR-10 and CIFAR100 datasets, LSUN is composed of high-resolution images; therefore, we evaluated ScoreGAN with LSUN to verify that the proposed framework can be performed with high-resolution images. In this experiment, ScoreGAN produces 128×128 resolution images.

The training process is the same as the previous experiments with the CIFAR datasets, while different training parameters were used; a learning rate of 5×10^{-5} was used for both the generator and discriminator, and the weights of the discriminator were updated two times for each update of the generator. Furthermore, the number of layers of the generator and discriminator was increased due to the resolution of the produced images. Since the resolution of the images is four times that of the CIFAR datasets, two additional residual modules were employed, which correspond to four additional convolutional layers for both the generator and discriminator.

Examples of the generated images by ScoreGAN are shown in Figure 5. The proposed model produced fine images for each category in the LSUN dataset. These results confirm that the proposed model can be applied to higher-resolution images than those in the CIFAR datasets, which demonstrates the generality of the performance of the proposed model. The result of the additional experiments signifies that the proposed model can be trained with various image datasets that have many image categories, such as CIFAR-100, as well as datasets with high-resolution images, such as LSUN.

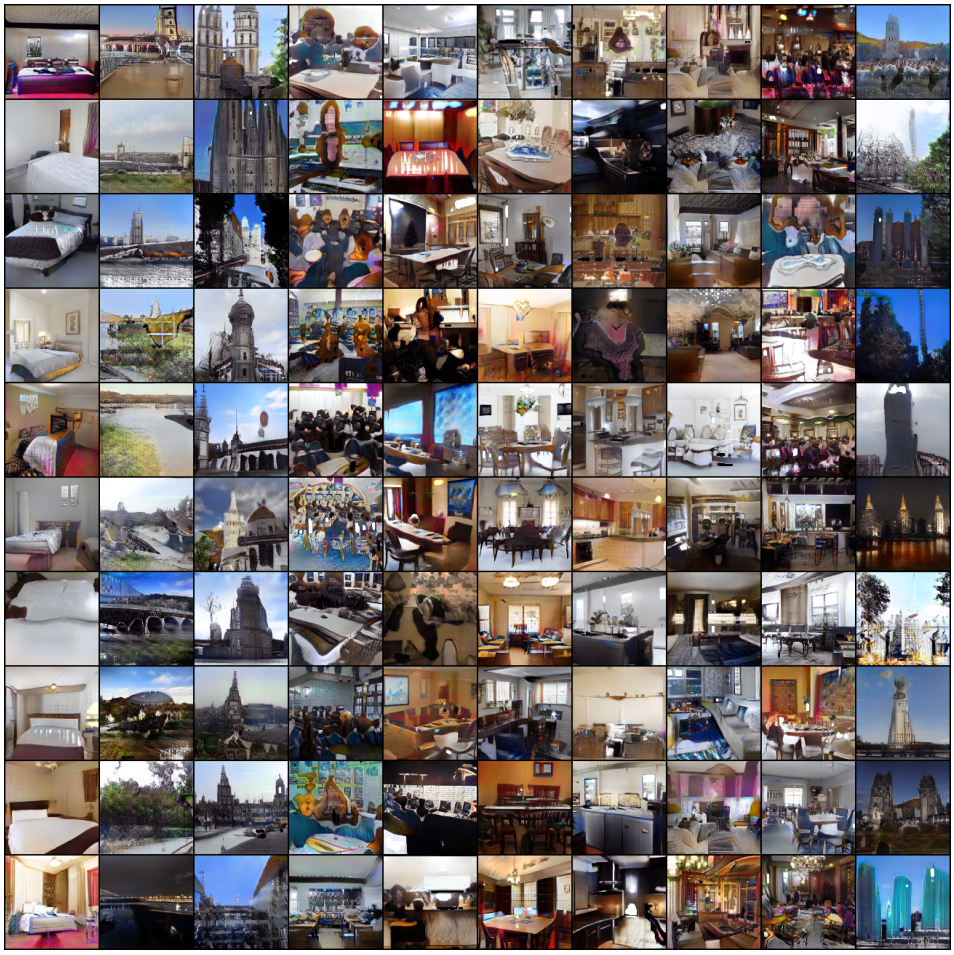

Figure 5. Random examples of the generated images by ScoreGAN with the LSUN dataset. The images are a 128 × 128 resolution. Each column represents each class in the LSUN dataset, i.e., bedroom, bridge, church outdoor, classroom, conference room, dining room, kitchen, living room, restaurant, and tower.

5. Conclusions

In this paper, the proposed ScoreGAN introduces an evaluator module that can be integrated with conventional GAN models. While it is known that the regular use of the Inception score to train a generator corresponds to making noise-like adversarial examples of the Inception network, we circumvented this problem by using the score as an auxiliary target and employing MobileNet instead of the Inception network. The proposed ScoreGAN was evaluated over the CIFAR-10 dataset and CIFAR-100 dataset. As a result, ScoreGAN demonstrated an Inception score of 10.36, which is the best score among the existing models. Furthermore, evaluated over the CIFAR-100 dataset in terms of the FID, ScoreGAN outperformed the other models, where the FID was 13.98.

Although the proposed evaluator is integrated with the ControlGAN architecture and demonstrated fine performance, it needs to be further investigated whether the evaluator module properly performs when it is additionally used for other GAN models. Since the evaluator module can be employed along with various GANs, the performance can be enhanced by adopting other GAN models. Furthermore, in this paper, only the Inception score is introduced to train the generator while the other metric to assess GANs, i.e., the FID, can be used as a score. Such a possibility to use the FID as a score should be further studied as well for future work.

Author Contributions: Conceptualization, M.L. and J.S.; methodology, M.L.; software, M.L.; validation, M.L.; formal analysis, M.L.; investigation, M.L. and J.S.; writing—original draft preparation, M.L.; writing—review and editing, M.L. and J.S.; supervision, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) grants funded by the Korean government (MSIT) (Nos. 2022R1A2C2004003 and 2021R1F1A1050977).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Neural Network Architectures of ScoreGAN for the CIFAR-100 Dataset

Table A1. Architecture of the neural network modules for the training of the CIFAR-100 dataset. The values in the brackets indicate the number of convolutional filters or nodes of the layers. Each ResBlock is composed of two convolutional layers. The difference between the architecture for the CIFAR-10 dataset is at the classifier, in which 256 filters are used in the last three ResBlocks.

Generator	Discriminator	Classifier
$Z \in \mathbb{R}^{128}$	$Z \in \mathbb{R}^{32\times32\times3}$	$Z \in \mathbb{R}^{32\times32\times3}$
Dense $(4 \times 4 \times 256)$	ResBlock Downsample (256)	ResBlock (32) $\times$ 3 ResBlock Downsample (32)
ResBlock Upsample (256)	ResBlock Downsample (256)	ResBlock (64) $\times$ 3 ResBlock Downsample (64)
ResBlock Upsample (256)	ResBlock (256)	ResBlock (128) $\times$ 3 ResBlock Downsample (128)
ResBlock Upsample (256)	ResBlock (256)	ResBlock (256) $\times$ 3
cBN; ReLU; Conv (3); Tanh	ReLU; Global Pool; Dense (1)	LN; ReLU; Global Pool; Dense (100)

References

1. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv* **2022**, arXiv:2204.06125.
2. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
3. Aggarwal, A.; Mittal, M.; Battineni, G. Generative adversarial network: An overview of theory and applications. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100004. [CrossRef]
4. Kim, W.; Kim, S.; Lee, M.; Seok, J. Inverse design of nanophotonic devices using generative adversarial networks. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105259. [CrossRef]
5. Park, M.; Lee, M.; Yu, S. HRGAN: A Generative Adversarial Network Producing Higher-Resolution Images than Training Sets. *Sensors* **2022**, *22*, 1435. [CrossRef]
6. Lee, M.; Tae, D.; Choi, J.H.; Jung, H.Y.; Seok, J. Improved recurrent generative adversarial networks with regularization techniques and a controllable framework. *Inf. Sci.* **2020**, *538*, 428–443.
7. Cai, Z.; Xiong, Z.; Xu, H.; Wang, P.; Li, W.; Pan, Y. Generative adversarial networks: A survey toward private and secure applications. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–38. [CrossRef]
8. Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
9. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410.
10. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2223–2232.
11. Kim, S.W.; Zhou, Y.; Philion, J.; Torralba, A.; Fidler, S. Learning to simulate dynamic environments with GameGAN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
12. Lee, M.; Seok, J. Estimation with uncertainty via conditional generative adversarial networks. *Sensors* **2021**, *21*, 6194. [CrossRef]
13. Yi, X.; Walia, E.; Babyn, P. Generative adversarial network in medical imaging: A review. *Med. Image Anal.* **2019**, *58*, 101552.
14. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 7354–7363.

15. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
16. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training GANs. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Barcelona, Spain, 5–10 December 2016; pp. 2234–2242.
17. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 6626–6637.
18. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
19. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2805–2824. [CrossRef]
20. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
21. Barratt, S.; Sharma, R. A note on the inception score. *arXiv* **2018**, arXiv:1801.01973.
22. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
23. Zhu, Y.; Wu, Y.; Olszewski, K.; Ren, J.; Tulyakov, S.; Yan, Y. Discrete contrastive diffusion for cross-modal and conditional generation. *arXiv* **2022**, arXiv:2206.07771.
24. Song, J.; Meng, C.; Ermon, S. Denoising Diffusion Implicit Models. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
25. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
26. Miyato, T.; Koyama, M. cGANs with projection discriminator. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
27. Ni, Y.; Song, D.; Zhang, X.; Wu, H.; Liao, L. CAGAN: Consistent adversarial training enhanced GANs. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; pp. 2588–2594.
28. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; pp. 2642–2651.
29. Lee, M.; Seok, J. Controllable generative adversarial network. *IEEE Access* **2019**, *7*, 28158–28169. [CrossRef]
30. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
31. Chen, H.Y.; Su, C.Y. An enhanced hybrid MobileNet. In Proceedings of the International Conference on Awareness Science and Technology (iCAST), Fukuoka, Japan, 19–21 September 2018; pp. 308–312.
32. Qin, Z.; Zhang, Z.; Chen, X.; Wang, C.; Peng, Y. FD-MobileNet: Improved MobileNet with a fast downsampling strategy. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1363–1367.
33. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein GANs. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5767–5777.
34. Lee, M.; Seok, J. Regularization methods for generative adversarial networks: An overview of recent studies. *arXiv* **2020**, arXiv:2005.09165.
35. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral normalization for generative adversarial networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
36. Lim, J.H.; Ye, J.C. Geometric GAN. *arXiv* **2017**, arXiv:1705.02894.
37. Dumoulin, V.; Shlens, J.; Kudlur, M. A learned representation for artistic style. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
38. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
39. Kavalerov, I.; Czaja, W.; Chellappa, R. cGANs with Multi-Hinge Loss. *arXiv* **2019**, arXiv:1912.04216.
40. Wang, D.; Liu, Q. Learning to draw samples: With application to amortized mle for generative adversarial learning. *arXiv* **2016**, arXiv:1611.01722.
41. Grinblat, G.L.; Uzal, L.C.; Granitto, P.M. Class-splitting generative adversarial networks. *arXiv* **2017**, arXiv:1709.07359.
42. Shmelkov, K.; Schmid, C.; Alahari, K. How good is my GAN? In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 213–229.
43. Tran, N.T.; Tran, V.H.; Nguyen, B.N.; Yang, L. Self-supervised GAN: Analysis and improvement with multi-class minimax game. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019; pp. 13232–13243.
44. Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv* **2015**, arXiv:1506.03365.