Supplementary Material for:
A Survey on Generative Adversarial Networks: Variants, Applications, and Training

ABDUL JABBAR, XI LI, and BOURAHLA OMAR, College of Computer Science, Zhejiang University, Hangzhou, China

1 GAN VARIANTS

Due to the popularity and success of GANs [1], many adjustments and variations over the original model of standard GANs have been proposed to improve training. Here, we discuss some of the modified versions of GANs. Additionally, we also provide GANs-variants analysis in terms of their evaluation metrics. Prior knowledge about the baseline model's evaluation metrics may be crucial before applying it successfully for a practical application.

1.1 Conditional GAN (C-GAN)

Conditional GAN (C-GAN) [2] is the conditional version of GAN. These networks can be constructed by merely feeding the extra auxiliary information (e.g., class label) extending the standard GAN into C-GAN. The generator of C-GAN takes the extra auxiliary information c (class label, text, or images) and a latent vector (z) so that it generates conditional real-looking data ($G(z|c)$). The C-GAN discriminator takes the extra auxiliary information c (class label, text, or images) and real data (x) to distinguish generator generated samples and real data. The C-GAN dictates the type of data generated by the generator through conditional variables, applied to the generator and the discriminator, which is impossible to control for standard GAN. The basic architecture of Conditional GAN (C-GAN) is shown in Figure 1. The C-GAN is a general framework that can generate high-quality images with numerous characteristics such as diverse smiley faces, wear glasses, and hair color with a simple condition. However, C-GAN has some limitations. It has a substantial restriction of possessing the labeled dataset, and its generated objects for each condition are dissimilar. They do not permit a similar object to be generated under different conditions. The C-GAN training objective is defined as:

$$\min_G \max_D V_{CGAN}(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log (D(x|c))] + \mathbb{E}_{z \sim p_z(z)}[\log (1 - D(G(z|c)))] \quad (1)$$

Where the input noise variables ($p_z(z)$) and conditional variable (c) are inputs in the generator. The real samples (x), generated samples, and conditional variable (c) are inputs in the discriminator.

1.2 Deep Convolutional GAN (DC-GAN)

A new class of **convolutional neural networks (CNN)** [19] is called **Deep Convolutional GAN (DC-GAN)** [3]. The DC-GAN was the first structure that practiced de-convolutional neural networks (de-CNN) structural design that significantly stabilizes GAN training. These frameworks consist of two networks: one network works as a CNN, called the generator network, and the

0360-0300/2021/10-ART157 $15.00
https://doi.org/10.1145/3463475

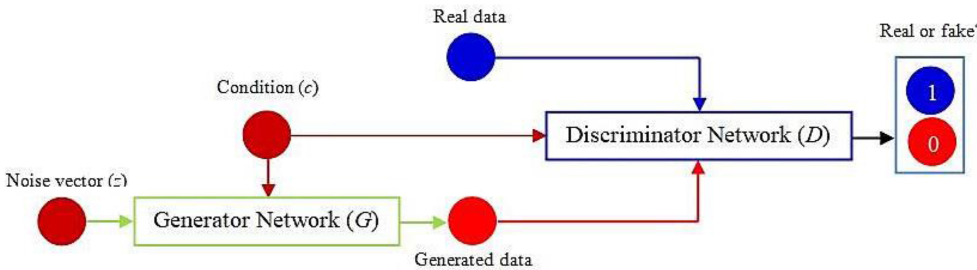

Fig. 1. The architecture of the Conditional Generative Adversarial Networks (C-GAN).

other network works as a de-CNN, called discriminator network. A newly proposed class of architectural constraints included in the CNN architecture is:

- Remove all levels of pooling layers with stride convolutions.
- Both G and D must use batch normalization [122].
- Use ReLU and Leaky-ReLU in both the networks (G and D), respectively.

The DC-GAN demonstrates a steady training procedure and excellent performance in superior quality sharp image generation tasks and offers impressive benchmark architecture for other GAN variants. This combined GAN and CNN approach can generate the best results, but removing the batch normalization layer [122] from the DC-GAN gives inferior quality results (samples have insufficient diversity).

1.3 Laplacian GAN (Lap-GAN)

A sequential image generation framework, **Laplacian GAN (Lap-GAN)** [4], was proposed by combining the C-GAN model [2] with the framework of the **Laplacian pyramid (LP)** [5]. The Lap-GAN requires the multi-scale generation process in which a series of the GAN generates particular levels of details of an image in an LP representation. Thus, the GAN at each generation step of the LP can be different. The LP was built from a **GP (Gaussian pyramid)** using up-sampling u (.) and down-sampling d (.) functions explained as:

Let $G(I) = [I_0, I_1 \ldots I_K]$ be the GP where $I_0 = I$ and I_K are k repetitive applications of d (.) to I. The coefficients h_k at each level (k) of the LP ($L(I)$) is given as:

$$h_k = L_k(I) = G_k(I) - u(G_{k+1}(I)) = I_\mathrm{k} - u(I_{k+1}) \tag{2}$$

The rebuilding of the LP coefficients $[h_1 \ldots h_K]$ through backward recurrence is:

$$I_k = h_k \ = u(I_{k+1}) + h_k \tag{3}$$

While training Lap-GAN, there is a set of generative models $[G_0 \ldots G_K]$, each of which captures the dissemination of coefficients h_k for diverse levels of the LP. Here, while reconstruction, the generative models are used to produce h_k's. Equation (3) is thus updated as follows:

$$\tilde{I}_k = u(\tilde{I}_{k+1}) + \tilde{h}_\mathrm{k} = u(\tilde{I}_{k+1}) + G_k(z_k, u(\tilde{I}_{k+1})) \tag{4}$$

Where at every level, a stochastic option is prepared to construct the coefficient $h_\mathbf{k}$ by the standard procedure or $G_{\boldsymbol{k}}$.

The Lap-GAN is the main adaptation for GAN that up-scales the low-resolution input image to an output image of higher resolution in a coarse-to-fine fashion, generating more photo-realistic images than standard GAN. Although Lap-GAN performs better than the other best GAN-variants in generating realistic and better quality images at higher resolution, its coarse-to-fine fashion approach is computationally expensive, and a very deep Lap-GAN does not converge quickly.

1.4 Information Maximizing GAN (Info-GAN)

Information Maximizing GAN (Info-GAN) [6] proposed an idea of the representation learning algorithm that can learn disentangled design in a wholly unsupervised way. The Info-GAN is a completely unsupervised framework built on top of GAN and disentangles both discrete and continuous latent factors, scales to complicated datasets, and requires no more training time than GAN. The Info-GAN training objective is defined as follows:

$$\min_{G}\max_{D} V_{InfoGAN}(D, G) = V(D, G) - \lambda . I(c; G(z, c)) \tag{5}$$

Where λ refers to a hyper-parameter, z refers to un-interpretable noise, c encodes the salient latent codes, and I is for the mutual information/shared information.

Compared to the standard GAN, the Info-GAN's uniqueness introduces a regularization term (I), as shown in the previous equation, capturing the shared information among the interpretable variables (c) and the generator output. The **Mutual Information (*I*)** is mathematically written as follows:

$$I(c; G(z, c)) = \text{Entropy}(c) - \text{Entropy}(c|G(z, c)) \tag{6}$$

The second entropy term needs admittance to the posterior ($c|G(z, c)$), which was approximated by D.

The unsupervised Info-GAN can also be perceived as a variant of C-GAN [2], making the image generation method more controllable. The outcome can be interpreted through the induction of mutual information. However, the use of the Info-GAN is better if datasets are not that complex because, in the case of a complex dataset, it gives inferior quality results. With the Info-GAN, it is also possible to modify the generated images' visual features, but it is not likely to stipulate which particular visual features we would like to vary because we can't forecast which training features will be learned. Similar to Info-GAN, **Semi-supervised Info-GAN (SS-InfoGAN**) [7] takes the advantages of supervised and unsupervised learning via optimizing the shared information between the synthesized data and the un-supervised latent code, and SS-InfoGAN can learn latent code representation from a smaller size unlabeled dataset more efficiently compared to the unsupervised Info-GAN.

1.5 Energy-Based GAN (EB-GAN)

Energy-Based GAN (EB-GAN) [8] is a variant of the GAN architecture which combined autoencoder and GAN frameworks where the discriminator works as an energy function which refers to the low energy to actual data and high energy to fake data instead of a routine GAN probability function that determines input as real or fake. The EB-GAN used two distinct losses for both G and D. When the EB-GAN generator was far from convergence, they got better in quality gradients and performance. The discriminator and the generator losses are formally given in Equation (7) and Equation (8), respectively:

$$L_D^{EBGAN}(\mathrm{x}, \mathrm{z}) = D(\mathrm{x}) + [\mathrm{m} - D(G(\mathrm{z}))]^+ \tag{7}$$

$$L_G^{EBGAN}(\mathrm{z}) = D(G(\mathrm{z})) \tag{8}$$

Where L_G, L_D, x, and $G(z)$ are the correspondence with the generator loss, discriminator loss, a data sample, and a generated sample. These equations also show that minimizing the generator loss L_G concerning G parameters is similar to maximizing the L_D concerning parameters D has the same minimum for the positive margin m.

The EB-GAN demonstrates better convergence power and produces realistic diverse images of excellent resolution by preventing the generator from providing similar samples through a regularization loss term. The EB-GAN shows better stable performance than standard GAN during training; however, fixed margin m phenomena used in EB-GAN makes it challenging to adjust to the varying dynamics of the G and D, which hurts the achievements of the discriminator in reconstructing the real samples because the energy value of the generated example varies close to the margin m.

1.6 Wasserstein GAN (W-GAN)

Wasserstein GAN (W-GAN) [9] proposed a substitute loss function derived through **Earth-Mover (EM)** or Wasserstein distance. Unlike the standard GAN cost function, where the discriminator works as a binary classifier, the W-GAN discriminator is used to fit the Wasserstein distance. The W-GAN is the most straight forward GAN to train using an alternate cost function that does not suffer from the vanishing gradient problem and partially removes the mode collapse training obstacle to stabilize the GAN training and get better results in terms of the "mode collapse" problem. But the W-GAN model may still produce low-quality samples and converge slowly due to weight clipping (i.e., weight clipping imposes a Lipschitz constraint on the critic network which represents the discriminator network), which can lead to an unwanted performance in some settings. In addition, W-GAN-based models have limited capability to model complex functions. The Earth-Mover (EM) distance is defined as follows:

$$W(p_r, p_g) = \inf_{\gamma \in \prod (p_r, p_g)} \mathbb{E}_{(x,y)\sim\gamma} [\|x - y\|] \tag{9}$$

Where $\Pi(p_r, p_g)$ refers to a collection of all mutual proportions, and the range of γ(x, y) are p_r (real data) and p_g (generated data).

As inf (infimum term) is intractable, so, Equation (9) can be reformulated in terms of Kantorovich-Rubinstein duality [11]:

$$W(p_r, p_\theta) = \sup_{\|f\|L \leq 1} \mathbb{E}_{x\sim p_r} [f(x)] - \mathbb{E}_{x\sim p_\theta} [f(x)] \tag{10}$$

Where sup (supremum term) is the least upper bound over all the 1-Lipschitz functions as shown in the above equation.

Similarly, **WGAN Gradient Penalty (WGAN-GP)** [10] framework further extended the idea of W-GAN with the introduction of a gradient penalty (GP) term on the discriminator for the implementation of the 1-Lipschitz condition. The WGAN-GP would improve GAN training stability, produce high-quality samples, and has much faster convergence power than WGAN-GP. WGAN-GP updated cost function as:

$$L_{WGAN-GP} = \mathbb{E}_{x_g \sim p_g}[D(x_g)] - \mathbb{E}_{x_r \sim p_g} [D(x_r)] + \lambda \, \mathbb{E}_{\hat{x} \sim p_{\hat{x}}} \left[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2 \right] \tag{11}$$

In Equation (11), x_r is the sample data drawn from p_r, x_g is the sample data drawn from p_g and $p_{\hat{x}}$ is the uniform distribution sampled between points in p_r and p_g.

1.7 Boundary Equilibrium GAN (BEGAN)

Boundary Equilibrium GAN (BE-GAN) [12] keep-up an equilibrium that manages the trade-off between variety and superiority. The main goal behind BE-GAN is to change the loss function. The Wasserstein distance between reconstruction loss of actual and synthesized images gives the real loss. In BE-GAN, the discriminator works during training as the autoencoder balances the optimizing process of G and D. The idea of making the discriminator as an autoencoder was first proposed in EB-GAN [8]. The BE-GAN objective is given as:

$$\left.\begin{aligned} L_D(x,\ z) &= \mathrm{D(x)} - k_\mathrm{t}\mathrm{D(G(z)} && \text{for } \theta_\mathrm{D} \\ L_G(z) &= \mathrm{D(G(z)} && \text{for } \theta_\mathrm{DG} \\ L_{k+1} &= k_\mathrm{t} + \alpha(\gamma \mathrm{D(x)} - \mathrm{D(G(z))}) && \text{for k training} \end{aligned}\right\} \tag{12}$$

Where $L_G,\ L_D$ represents the losses of the generator, the discriminator, L(x), L(G(zD)) represents the auto-encoder L1 loss of real, fake data, and equilibrium hyper-parameter "γ" respectively.

The BE-GAN uses Wasserstein distance instead of JS divergence to poise the generator's training and the discriminator. It is stable, fast, avoids over-fitting, and is resilient to parameter changes. The BE-GAN method also adds a new estimated convergence measure to stabilize human faces' training and generation with visually pleasing quality. The BE-GAN has made some significant performance in measuring convergence and image quality. But some shortages need to be enhanced, such as the trade-off between the variety and the images' quality. One more fundamental problem with BE-GAN's generator network is that it does not discover the low-probability features well.

1.8 Progressive-Growing GAN (PG-GAN)

The **Progressively-Growing GAN (PG-GAN)** [13] proposed a multi-scale GAN-based architecture where both *G* and *D* start their training with the low-resolution image (e.g., 4×4), gradually increase the model depth by adding up the new layers to both *G* and *D* during the training process, and ends-up with the generation of large scale sharp image (e.g., 1024×1024). The basic idea behind the PG-GAN is to grow both *G* and *D* in synchrony, i.e., starting from a low-resolution image (e.g., 4×4), duos the resolution of the generated image (e.g., 8×8) with the addition of new layers to both networks, and ends-up with the generation of higher resolution image (e.g., 1024×1024) as the training progresses.

The Progressive Growing GAN grows progressively and has been highly flourishing to improve quality, increase stability, and variation as compared to equivalent non-progressive GANs. The PG-GAN has additional benefits such as a set of first layers that converge quickly, only a few layers at a time are trained from scratch, and the training time is reduced significantly. Although the PG-GAN performance is excellent, it is still not satisfied with the mode collapse dilemma, i.e., the generator of the PG-GAN generates similar samples due to unbalance training of the generator network and discriminator network.

1.9 BigGAN

Big GAN (BigGAN) [14] is one of the best models due to its unprecedented, large scale, indistinguishable, and high-quality image generation capacity. BigGAN-a deep learning model- can train bigger neural networks with even more parameters; create a more extremely detailed image with remarkable performance. BigGAN have some essential properties such as it provides exerts control over the outputs, provides interpolation phenomena between images, which means that if there are two images, it can compute the intermediate image between them and offers the best inception score (IS), i.e., the best of earlier works had an inception-score (IS) around 50. The IS of the BigGAN technique is not less than 166, closer to real images, which would score around 233.

Although the performance of the BigGAN is exceptional in large and high-fidelity diverse image generation, with random sampling, the diversity of the generated image is much less than a real image of the same size. The model also has limited data augmentation ability on large-scale datasets such as ImageNet. One more fundamental problem with the BigGAN model is that it can't repeat the outcomes from scratch without sufficient data. A proposed extension to BigGAN called

Bi-Directional BigGAN (BigBiGAN) [15] improves the un-conditional image generation and representation learning capacity of the model, such as increased **freshet inception distance (FID)** and **inception-score (IS)** accuracy score over the baseline BigGAN [14] model for un-conditional results.

1.10 Style-Based Generator Architecture for GAN (StyleGAN)

Although PG-GAN [13] generates a high-quality image, its ability to control the generated image's specific features is minimal. To overcome this issue, **Style-Based Generator Architecture for GAN (StyleGAN)** [16] redesigns the architecture of the generator network, allows it to monitor the image synthesis through scale-specific amendments to the styles without compromising the generated image quality but increases it significantly utilizing PG-GAN. In fact, StyleGAN introduced an upgraded version of PG-GAN, which only focused on the generator network to control the co-relation between input features. StyleGAN divides the input features into three types, such as (I) coarse features-pose, hair, face, shape, (II) medium features - facial features, eyes, and (III) fine scheme. The StyleGAN is famous for its un-conventional GAN architecture, such as the use of a mapping network that first transforms the latent input code into inter-mediate latent code where affine transformation then produce styles that control the layers of the synthesis network through **Adaptive Instance Normalization (AdaIN)** [17] that scales the normalized input with style spatial statistics, and the PG-GAN [13] has been extremely flourishing in stabilizing GAN training.

Despite StyleGAN's excellent success in improving GAN's ability to have reasonable control over the generated images instead of focusing on producing more realistic-looking images. It also has some native artifacts, such as blob-shaped artifacts that resemble water droplets due to instance layer normalization and phase artifacts due to progressive growing phenomena that need to be removed. The StyleGAN2 [18] comes with various improvements to image quality, efficiency, diversity, and disentanglement, and the results are incredibly improved. The StyleGAN2 simply redesigns the normalization used in the generator of the StyleGAN [16], which removes the artifacts such as blob-shaped artifacts that resemble water droplets. The StyleGAN2 achieves excellent results in face image synthesis than the StyleGAN.

1.11 Comparative Analysis

This section provides GANs-variants analysis in terms of their evaluation metrics. Prior knowledge about the baseline model's evaluation metrics may be crucial before applying it successfully for a practical application. Table 1 summarizes the merits and demerits in tabular form for a better view of all the GAN variants discussed in this survey. The literature review shows that **Generative Adversarial Networks (GAN)** [1] have excellent performance in the generation of natural-looking realistic samples. GAN supports supervised, semi-supervised, and unsupervised learning approaches. The GAN uses the log-likelihood parameter for performance evaluation. **Conditional Generative Adversarial Networks (C-GAN)** [2] can control the generation of the image with its conditional variable applied on G and D. The purpose of any C-GAN-based model is to minimize the cost function for the generator and maximize the cost function for the discriminator concerning the applied conditional parameter. The C-GAN uses a log-likelihood parameter for performance evaluation. **Deep Convolutional Conditional Generative Adversarial Networks (DC-GAN)** [3] architecture demonstrates the steady training process and the superior quality of sharp images. Accuracy and error rate are performance evaluation metrics for the DC-GAN-based models.

Boundary Equilibrium GAN (BEGAN) [12] model balances both the networks (G and D) and adds a new estimated convergence measure. The BE-GAN method uses Wasserstein distance instead of JS divergence to stabilize human faces' training and generation process with high

Table 1. A Summary of the GAN-variant's Merits and Demerits

Model	Merits	Demerits
CGAN [2]	1. Minimize value function for G and maximize for D conditioned on extra information, which controls the data generation process. 2. Prevent mode collapse and produce high-quality images.	1. It gives preferably better performance for the only labeled dataset.
DCGAN [3]	1. Learn hierarchy of representations from object parts to scenes in both G and D. 2. More steady in terms of generating higher quality samples and training.	1. The misclassification rate is higher than other GAN-based models. 2. Little architectural changes may cause inferior quality and diversity in the generated images.
LapGAN [4]	1. Unsupervised learning technique. 2. Generation of images in coarse-to-fine fashion. 3. Improve image quality.	1. It has a higher computational cost. 2. A very deep LapGAN does not converge easily.
InfoGAN [6]	1. Additionally, it learns latent variables without labels in the data. 2. Performs well in a case where labels are missing. 3. Easy to train.	1. It gives better performance only when data is not very complex and small in size. 2. It can only memorize a sparse dataset if it is not very complex.
EBGAN [8]	1. EBGAN has more stable behavior than original GANs during training. 2. Solves the mode collapse problems.	1. Fixed value (margin m) phenomena hurt the performance of the discriminator in-reconstructing the real samples.
WGAN [9]	1. Train the discriminator until convergence, leading to higher quality samples generated by the generator. 2. Enables stable training for wide of GAN structures, without hyper-parameter tuning. 3. Solves the vanishing gradient and mode collapse problems.	1. A significant drawback is that large weight clipping incurs pathological behavior that contributes to slow convergence and causes a vanishing gradient. 2. Weight clipping reduces the capacity of the model and limits the capability to model complex functions.
BEGAN [12]	1. Introduced new loss functions by using auto-encoder as the discriminator. 2. It does simultaneous training of G and D in an adversarial way in each time step. 3. It balances the G and D capability using an equilibrium that can be adjusted for the trade-off between diversity and quality. 4. The equilibrium approach enables the network to produce visually pleasing images.	1. It cannot discover the low-probability features well. 2. The trade-off between diversity and quality is a big issue.
PGGAN [13]	1. Improves image quality, and variation. 2. It can discover the low-probability features well.	1. It has a mode collapse problem due to the un-balance training of both the networks.
BigGAN [14]	1. Produce high-quality images. 2. Increase the diversity and fidelity of the generated samples by increasing the batch size and using the "truncation trick".	1. It has a higher computational cost. 2. It has limited data augmentation ability. 3. If you do not have enough data, it can be challenging to replicate results from scratch.
StyleGAN [16]	1. Control data generation process very well through interpolation mechanism. 2. Produce high-quality images.	1. Its generated samples have some characteristics artifacts like blob facts.

quality. The BE-GAN uses the Inception-Score for performance evaluation metrics. **Laplacian GAN (Lap-GAN)** [4] is the GAN's main adaptation an up-scale LR input image to HR outputs image. The log-likelihood and human decisions are performance evaluation metrics used for the Lap-GAN-based models. **Energy-Based GAN (EB-GAN)** [8] demonstrates better convergence power and the capacity to produce realistic HR images. The Inception-Score is the performance assessment metric used for the EB-GAN-based generative models. **Wasserstein GAN (W-GAN)** [9] is the simpler and most stable model to train. W-GAN is similar to DC-GAN in generating images. Still, if you remove the batch normalization [122] from DC-GAN architecture, it is inferior in quality, but W-GAN does an excellent job without batch normalization. Inception-Score for performance evaluation metrics for W-GAN uses the based generative models.

Information Maximizing GAN (Info-GAN) [6] model learns the disentangled illustration of features by maximizing the shared information between observations and noise variables. The information metric and representation learning are performance evaluation metrics used for the Info-GAN-based generative models. **Progressive-Growing GAN (PG-GAN)** [13] grows progressively, and has been extremely flourishing in stabilizing GAN training. The multi-scale structural similarity and the sliced Wasserstein distance are performance evaluation metrics used for the PG-GAN-based models. BigGAN [14], is one of the current best models due to its unprecedented, large-scale, indistinguishable, and high-quality image generation capacity. The Inception-Score and the Frechet Inception Distance are the performance assessment metrics used for the BigGAN-based generative models. **Style-Based Generator Architecture for GAN (StyleGAN)** [16] improves GAN's ability to have reasonable control over the generated image instead of focusing on generating more realistic-looking images. The Frechet Inception Distance is a performance assessment metric used for the StyleGAN.

2 ANALYSIS OF GAN APPLICATIONS MERITS AND DEMERITS

This section provides a much-needed in-depth analysis of GAN applications merits and demerits discussed in this survey, along with Table 2, which summarizes each GAN-based application's merits and demerits in tabular form to better view the generative adversarial network's outstanding data generation capability.

The handwritten font generation approaches [21–27] can automatically generate realistic-looking handwritten characters indistinguishable from human-written characters. But these approaches cannot perform well with a modest size dataset; they require a large dataset for training. Anime character generation applications [28, 29] generate realistic-looking animation characters without professional artistic skills. Anime character generation has vast applications, including animation production, game development, and anime characters' auto-colorization. However, they fail to generate high-quality results and often produce blurred and distorted anime characters. The blending approaches [30, 31] can generate realistic-looking blended images without manual effort. Still, these approaches have difficulty in blending objects with diverse poses. GANs have been successfully applied to perform image-painting tasks [32–34] that could bring out authentic impaired images into existence. However, in these networks, the image inpainting job is frequently resolved within a similar domain, e.g., the input image and output image are both RGB images. The face aging methods [35–39] have recently accomplished excessive facial aging success to forecast future looks and approximate previous looks. All these models allow facial aging from one age group to another. But, there is still much space to enhance the efficiency of facial aging approaches in aging precision.

The text-to-image generating approaches [40–44] have shown promising results in many uses, including computer-aided content creation and photo editing. Still, the achievement of existing methods has been mainly restricted to simple datasets such as birds and flowers. The generation

Table 2. A Summary of GAN-applications Merits and Demerits.

GAN Applications	Merits	De-merits
Handwritten font generation		
Zi2zi [21]	Automatic Chinese calligraphy font generation technique based on paired images, which considers each character as a whole and learns to transform between fonts.	The target font is restricted to many characters, such as Japanese and Chinese; thus, it is hard to apply to alphabets that contain few letters.
DenseNet-CycleGAN [22]	Show a successful use of domain adaptation by learning a mapping between two image distributions	It is restricted to a handful of Chinese characters only.
LS-CGAN [23]	Suggested a stroke-based font generation technique where two trained styles can be interpolated by monitoring a weight.	It does not deal with the level of abstraction of semantic properties that might be more valuable for humans.
GlyphGAN [24]	Employs only abstracted inputs as vectors, thus allowing the generation of fonts not seen in the training image.	It is hard to keep legibility and style steadiness in this approach
1. Anime characters generation		
Anime character GAN [28]	Generates artificial anime-style faces by joining a given tag with random noise.	Needs training label for specified and unspecified factors of interest or is limited to binary attributes.
PS-GAN [29]	Generates full-body HR anime character images by progressively growing the resolution of structural conditions and generated images during training.	It fails to generate diverse samples due to unbalance training of G and D.
3. Image blending		
GP-GAN [30]	Generates high-resolution well-blended images while protective caught high-resolution facts.	It produces unrealistic illumination and boundary effects.
GCC-GAN [31]	First, predict the tri-map, and then carried out the matting process.	Separates the process of segmentation and matting in different stages, which may generate erroneous segmentation results that mislead the matting step.
4. Image inpainting		
Ex-GAN [32]	Produce high-quality and personalized inpainting results by utilizing a reference image as exemplar information.	Edit eyes only and requires reference images of the same identity.
CA-GAN [33]	Copy-paste similar feature patches from visible regions to the missing regions.	The absence of fine texture details and pixels is not consistent with the background.
PG-GAN [34]	Bring out very photorealistic impaired images.	Manual generation of masks is time-consuming.
5. Face aging synthesis		
Age-CGAN [35]	Generate high-quality and incredibly realistic results.	It can't perfectly preserve the original age's face identity.
CAAE [36]	Generates realistic face images and offers more robustness against facial expression pose variation and occlusion difficulties.	The success is related to the availability of an extensive database with different ages, so performance is not realistic for a small amount of data.
IP-CGAN [37]	Utilize pre-trained networks to preserve identity & achieve substantial identity preservation results.	The main disadvantage comes from the nature of deterministic (uncontrollable) one-to-one mapping.
Wavelet-GANs [38, 39]	Achieve better aging performance than conventional methods such as physical and prototype-based methods.	Have limitations in discovering the disentangled factors in facial aging.

(Continued)

Table 2. Continued

GAN Applications	Merits	De-merits
6. Text-to-image synthesis		
Attn-GAN [40]	It uses a word-level visual-semantic that fundamentally relies on a sentence vector to generate images.	Using word-level attention alone does not ensure global semantic consistency due to the diversity between text and image modalities.
Stack-GAN [41]	Shows higher output diversity than other text-to-image translation models.	Handling multiple networks is a difficult task.
AC-GAN [42]	Creates high-resolution, sharper, and better quality images for text-to-image synthesis.	As the number of labels increases, the model tends to generate near-similar images for most classes.
TAC-GAN [43]	Uses a text description condition instead of a class label condition for text-to-image synthesis.	Success limited to simple datasets.
SIS-GAN [44]	Successfully manipulates the images with semantically different text descriptions generated from the original text.	Performance is limited due to a less effective text-image concatenation method.
7. Human pose synthesis		
PG^2 [45]	Allows person images to be synthesized in random poses, depending on the person's image and a novel pose.	It can't provide more controllable and varying person images conditioning on attribute and pose.
Deformable-GAN [46]	Generate a person's image conditioned on appearance and poses information.	Because of inadequate appearance data in a single source picture, this approach faces difficulties.
8. Stenographic applications		
S-GAN [47], and SS-GAN [48]	They produce more steganalysis-secure message embedding to enhance speed, training stability, and image quality via the steganography model.	The steganalyser model is so simple that the generated cover images produced by frameworks are not safe enough for steganography tasks.
Stegano-GAN [49]	A robust method of image steganography and found to be effective when tested on multiple data sets.	Applied only on lossless stego images; otherwise, confidential data portions get lost.
9. Image manipulation		
IGAN [50]	Provide a user's friendly interface to change the image interactively on a natural image manifold and vision.	It can only be applied in SR in a highly constrained setting for the face's task.
TA-GAN [51]	A text-adaptive discriminator that can provide the generator with detailed word-level training input has been implemented.	It has limited performance due to a coarse sentence condition and a less effective text-to-image concatenation method.
IAN [52]	Conducts image editing precisely according to the purpose of the recipient.	This style of work does not use the information from the labels to train G.
Att-GAN [53]	Performs facial attribute transfer based on the target attributes using an encoder-decoder net and treats the attribute information as part of the latent representation.	Fail to generate plausible results when the context and background pixels are absent.
D-GAN [54]	Presents a new framework to visualize and quantify individual units' role in a network by comparing each unit's activity with a range of human interpretable pattern-matching tasks such as detecting object classes.	It mainly focused on characterizing existing networks' interpretability and is less effective at connecting neural networks with traditional iterative algorithms and motivating. Novel network architectures.
10. Saliency prediction		
Sal-GAN [55]	This is the first state-of-the-art method that can generate saliency map prediction that resembles the ground truth.	It focuses on saliency prediction on a single image and needs ground truth to do supervised learning.
SalCapsule-CGAN [55]	It can predict saliency maps indistinguishable from the actual saliency maps.	Can't detect salient objects in a noisy scene.
DSAL-GAN [57]	Capable of handling noisy images holistically.	Handling multiple networks is a difficult task

(Continued)

Table 2. Continued

GAN Applications	Merits	De-merits
11. Object detection		
Se-GAN [58]	Segment not only invisible object parts but also reveal their appearance.	The lack of available data makes image segmentation a challenging endeavor, even today.
P-GAN [59]	It brings extraordinary progress in natural image generation with diverse conditions.	Requires features to be extracted from a pre-trained deep network, which is time and space-consuming.
MT-GAN [60]	Generates small high-resolution objects to improve multi-class detection performance.	Its RoI SR cannot take the context information into account since it focuses only on the ROIs.
GAN-DO [61]	A general framework that leads to robust object detection for images with varying quality.	Achieves the best performance in terms of average precision across all image quality types except defocus blur.
12. 3 D image synthesis		
3D-GAN [62]	Generate objects in a 3D voxel form by exploring the use of the VAE-GAN hybrid model.	The computational and spatial complexities of using such voxelized representations in standard encoder-decoder networks significantly limit the output resolution.
Pr-GAN [63]	The framework has shown success in learning 3D volumetric representations from 2D observations based on similar principles of projective geometry.	This method focuses on rigid objects, and they do not apply to dynamic and articulated human poses.
3D-CGAN [64]	Presents a model able to perform rotations of volumes in the 3D space.	The volumetric representation inevitably leads to (high) computational consumption and resolution limitation.
13. Medical applications		
Medical applications [67, 74]	Performs medical imaging, tumour detection, disease prediction, artifacts removal in X-rays, magnetic resonance imaging, and ultrasound image tasks.	It is always difficult to obtain medical datasets due to data privacy.
14. Facial makeup transfer		
BeautyGAN [75]	A framework with dual input and output for makeup transfer and removal simultaneously	Often fails on transferring in-the-wild images and cannot modify transfer partially and y precisely.
PairedCycle-GAN [76]	Introduced a makeup transfer and removal system by coupling two separate networks (GANs) together.	This approach does not work as well on severe makeup styles unseen during training.
DM-TGAN [77]	It transfers the makeup style from one image to another, but it also controls the strength on which it is applied.	Struggle to handle lower resolution data encountered in the real world.
15. Facial landmark detection		
SAN [78]	Successfully handle the significant intrinsic discrepancy of image styles.	It can't be applied to high-resolution jobs, such as segmentation.
Exposing GAN [79]	Tries to determine that facial landmarks of the generated face image have different configurations from the real human face image due to the lack of global constraints.	Inconsistent results.
Image super-resolution		
SRGAN [80]	It takes an image of low-resolution and induces an image of higher resolution through 4x up-scaling factors.	Generated texture information is not sufficiently real due to noise.
ESGAN [81]	Achieves more important visual quality results with more photo-realistic-looking textures than the SR-GAN.	Its architectural designs are very complicated.
SR-GDAN [82]	Solves the noise before the SR problem for the generation of the noise-free high-resolution images. As a result, it shows better performance than SR-GAN and ESR-GAN models.	Joint training of two networks is a challenging task.

(Continued)

Table 2. Continued

GAN Applications	Merits	De-merits
16. Texture synthesis		
M-GAN [84]	Efficiently synthesize texture by being trained on an unpaired collection of natural images and paintings.	Require many repetitions and a distinct run for each output image.
S-GAN [86]	Apply adversarial training at the pixel level to encourage the generated results to be indistinguishable from the real texture.	Limited to a single texture, they were trained on.
PS-GAN [87]	Learns to synthesize a collection of textures present in a single photograph, making it more general and applicable to texture interpolation.	It can suffer from mode collapse.
17. Sketch synthesis		
T-GAN [88]	Allows users to control object texture by dragging one or more example textures onto sketched objects, and the network realistically applies these textures to the indicated objects.	It is not able to handle complex scenes texture.
Sketchy-GAN [91]	Generates excellent results for single-class images.	It cannot generate multi-class images.
CA-GAN [92]	Improves the realism of the synthesized face photos and sketches by proposing a compositional reconstruction loss and using stacked CA-GANs.	It fails to synthesize realistic sketch textures and introduces many artifacts in the resulting images.
Image-to-image translation		
Pix2Pix [20]	Common framework for all automatic problems defining as the approach of translating one possible instance of an image into another by giving sufficient training data.	It has flaws in generating remote sensing images with high spatial resolution and cannot meet the needs for equivalent applications.
PAN [93]	Introduced an adaptive perceptual loss that automatically discovered the difference between the generated image and the ground truth with higher layer-based abstractions from deep networks.	It is not dedicated to unpaired image translation; comparing the mean difference between two domains would discard the characteristics of the generated image, leading to poor performance.
Cycle-GAN [94], Disco-GAN [95], Dual-GAN [96]	All adopt the idea of cyclic consistency, use unpaired data to train mapping from X space to Y, & vice versa.	All methods perform the image-to-image translation in only two domains (i.e., x to y, and y to x).
StarGAN [97]	Performs image translation among multi-domains by learning one model.	It cannot preserve the irrelevant regions of the face.
UNIT [98]	Performs I2I from one domain to another without any consequential images in the training dataset.	It fails to generate diverse results.
MUNIT [99]	Take single input and can generate multimode outputs.	Limited performance by the quality of I2I translation.
DRIT [100]	Generate diverse samples with un-paired training data.	It fails when the source and target domain characteristics vary considerably.
18. Face frontal view generation		
Face frontal view generation applications [102-106]	The face frontal view generation approaches can help the face recognition system (FRS) or guessing one's identity by producing high-quality frontal face images because any FRS's efficiency or identity may often degrade in the presence of a non-frontal face.	However, in these applications, the loss of too much image detail may often happen during non-frontal face to frontal face function.
19. Speech and language synthesis		
Rank-GAN [107] and VAW-GAN [108]	Generates the sequential data in the natural language processing field, like fake text or speech data.	However, these applications cannot generate fake text or speech data of better quality.

(Continued)

Table 2. Continued

GAN Applications	Merits	De-merits
20. Music generation applications		
C-RNN-GAN [109]	Generates continuous sequential data, i.e., precisely gaining the full sequences of music.	Its results are highly poor since it does not reflect discrete property of music elements.
Seq-GAN [111]	Trains GAN with Reinforce and Monte-Carlo search to generate word sequences. The estimates come with a high variance. Moreover, SeqGAN focuses on sampling one word (i.e. one-hot) at each timestep,	The word embedding with arbitrary projection does not efficiently capture relative harmony and consonance of each word.
OR-GAN [112]	Improves the diversity of the generated samples while maintaining the likeliness of the data distribution.	It can only be applied to sequential data, i.e., cannot be applied to non-sequential data.
21. Video applications		
VGAN [113]	Generates a video by mapping a sequence of random vectors to a sequence of video frames.	Generated video usually contains some flaws in temporal aspects.
MoCoGAN [114]	Separates the content from the movements to provide more control over the components.	The resolution of the generated video is low.
DRNET [115]	Separately capture the motion and semantic content.	Only operates on short video sequences.
22. Climate & earth science domain		
Climate & earth science applications [117-121]	These GAN-based methods offer accurate instantaneous earlier precautionary measures, which are directly related to human life.	However, the increasingly widespread practical applications warrant a more systematic evaluation of the possibilities and limitations of neural networks for the simulation of complex dynamical systems.

of real-world, complex images such as **Microsoft Common Objects in Context (MS-COCO)** remains an open challenge. The human pose synthesis methods [45, 46] support the image generation task via domain, person-specific knowledge, improved quality, and manageable image generation results. These methods built on the encoder-decoder architecture lack in-seeing the vital appearance and shape misalignments, frequently leading to unproductive generated images. The stenographic applications [47–49] provide an excellent way to hide secret information, e.g., a document, an image, or a video, within non-secret information. However, we are not aware of any text-stego system based on neural networks. The image manipulation approaches [50–54] have shown promising results by allowing users to describe visual concepts; it provides a natural and flexible interface for conditioning image generation. However, these manipulation applications somehow require professional skills and knowledge. The visual saliency prediction methods [55–57] get accurate results on saliency detection and successfully describe the region of interest in an image or video that attracts human attention. But these approaches often fail to detect ambiguous foreground objects, such as holistic structure images.

The small-object detection methods [58–61] detect the small objects successfully and reduce the blurriness and glaring errors in the results. Object detection methods have many practical applications, including surveillance, security, image retrieval, and advanced driver assistance systems. Although these small object detection methods perform well, they are far from good results in noisy and foggy visual conditions for practical applications. 3D image synthesis [62–64] can construct 3D (3 dimensional) object from 2D (2 dimensional) images. It has many applications, including computer graphics, computer-aided geometric design, medical imaging, computer animation, etc. Despite 3D image synthesis approaches, it faces many drawbacks, including high computational cost and low-resolution 3D constructed images. The potential effects of GAN applications [67–74] are important in the field of healthcare systems, such as medical imaging, tumor detection, disease prediction, artifacts removal in X-rays, magnetic resonance imaging, and ultrasound images. However, it is always difficult for GAN applications to obtain medical datasets due to data privacy issues. The facial makeup transfer applications [75–77] provide a ubiquitous way to improve one's facial appearance by shifting the makeup style from one face image to another while preserving the face image identity. However, these makeup transfer approaches require numerous samples with different makeup styles provided in the training data to make the algorithm more robust; otherwise, it will increase the difficulty of accurate face recognition.

The landmark detection techniques [78, 79] successfully detect the vital key points that help identify the facial images' discriminating regions. Though the existing landmark detection strategies have achieved significant performance improvements for facial landmark detection, enormous space still exists for enhancement, especially if some non-face object partially occludes facial image. The performance of image super-resolution algorithms [80–83] improves the visual quality results of low-quality results, which have significant importance, such as high-quality medical images. Although the current super-resolution GAN algorithms' performance is helpful in generating high-quality realistic images, they can only add little details to low-resolution images and cannot correct large defects. The other most significant challenge facing super-resolution algorithms is that it lacks realistic datasets, i.e., the noisy LR (low-resolution) image, and the corresponding HR (high-resolution) image. The texture synthesis methods [84–87] synthesize high-resolution textures and meaningfully increase the quality of generated samples. Even if these approaches have a strong capability to produce reasonable samples, they have restricted real-world applications due to the random and uncontrollable generated samples.

The GAN shows excellent sketch-to-image translation applications [88–92] by producing remarkable and realistic-looking images from rough sketches. Although the methods mentioned above have shown impressive results, they showed limited, ineffective, and inefficient multi-instance sketch generation performance. Image-to-image translation techniques [93–101] have been successfully applied for image translation and have many real-world applications, such as super-resolution, style transfer, object transfiguration, and medical imaging. However, these image-to-image translation approaches usually suffer from the mode collapse issue, regardless of whether it uses a single input or operates on multiple modes. The face frontal view generation approaches [102–106] can help the **face recognition system (FRS)** or guessing one's identity by producing high-quality frontal face images because any FRS's efficiency or identity may often degrade in the presence of a non-frontal face. However, in these applications, the loss of too much image detail may often happen during non-frontal face to frontal face function.

The language and speech synthesis approaches [107, 108] have synthesized the sequential data in the natural language processing field. However, these GAN's current applications cannot generate better text data or speech data because GAN is most often defined as continuous real value data. The music generation approaches [109–112] can produce new and different compositions for musicians. However, GANs' performance on music data is still facing some artifacts. Therefore, the generation of artifacts-free sequential data is a promising future research direction. The video applications [114–116] can successfully generate small-duration videos (e.g., future frame prediction) and perform video recognition tasks (e.g., action classification). However, current video applications require a large size memory and cannot generate videos of longer duration. Also, most of the current video applications' models have a static nature. Creating a dynamics nature model is challenging because there are many ways that scenes and objects can change. The GAN-based methods [117–121] for the climate & earth science domain offer accurate instantaneous earlier precautionary measures directly related to human life. However, the increasingly widespread practical applications warrant a more systematic evaluation of neural networks' possibilities and limitations for the simulation of complex dynamical systems.

3 NORMALIZATION TECHNIQUES FOR GAN TRAINING

GANs are computationally expensive due to their more considerable training time. The training time of GAN can be reduced with the help of normalizing operations. Different normalization techniques are proposed to minimize training time, which improves the stability of GAN during training. Here, we review different normalization techniques that can stabilize GAN model training.

3.1 Batch Normalization (BN)

Batch normalization (BN) [122] is a crucial technique to get deeper models to function without collapse. It is a pre-processing step applied to the intermediate layers (hidden layers) that reduces the internal covariate shift and avoids mode collapse by normalizing each mini-batch of data utilizing mean (μ) and variance (σ).

Let suppose a mini-batch of input data samples $\{x_1, x_2, \ldots, x_m\}$, and we compute the mean μ and variance σ^2 of the layer's input such as follows:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{13}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2 \tag{14}$$

and then replace each x_i with its normalized version such as:

$$\widehat{x_i} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{15}$$

Where ϵ is the constant factor added for numerical stability in Equation (15), which shows the result of BN for data samples (x_i). Better normalization provides better accuracy and regular training.

3.2 Virtual Batch Normalization (VBN)

Virtual batch normalization (VBN) [123] is an extension of the famous batch normalization (BN) technique that normalizes input examples by using the statistical data of many other inputs in the same mini-batch. To keep away from this issue where several other input data statistics are being used, the VBN normalizes each example based on collected data from a batch of references chosen on one occasion and pre-determined for the training. The reference batch normalizes statistical data only. The VBN is computationally costly, so it is mostly used only in the generator.

3.3 Layer Normalization (LN)

Layer normalization (LN) [124] normalizes the features across each example instead of normalizing the samples across mini-batches of the training datasets. The LN is advantageous over the famous BN [122] method like LN reduces the training time considerably better than the other normalization technique.

Let us suppose for input data x_i of dimension m, we compute the mean μ_i and variance σ_i^2 such as:

$$\mu_i = \frac{1}{m}\sum_{i=1}^{m} x_{ij} \tag{16}$$

$$\sigma_i^2 = \frac{1}{m}\sum_{i=1}^{m}(x_{ij} - \mu_i)^2 \tag{17}$$

and then replace each component x_{ij} with its normalized version such as:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_i}{\sqrt{\sigma_i^2 + \varepsilon}} \tag{18}$$

Equation (18) shows the result of LN, where i represents batch and j represents features. $x_{i,\mathrm{j}}$ is the i, j-th element of the input data samples. ϵ is for numerical stability if the denominator becomes zero by chance and is an arbitrarily small constant.

3.4 Weight Normalization (WN)

Weight normalization (WN) [125] has quite distinct advantages over other normalization techniques like WN, which has smaller calculation costs and easy implementation than other famous normalization techniques. Instead of normalizing the mini-batches directly, WN normalizes each layer's weights to optimize the neural network's weights. WN's computational cost is free from batch-size training data and has stable training accuracy across a broad span of batch size.

Let suppose; we think of a neural network where the computational cost of every neuron defined as:

$$y = \emptyset\ (\mathrm{w.\ x} + b) \tag{19}$$

By fixing k-dimensional vector v of weight vector w, we have new re-parameterization weight term, i.e., $\|\ \mathrm{w}\ \| = g$ (weight normalization).

$$\mathrm{w} = \frac{g}{||\mathrm{v}||}\ \mathrm{v} \tag{20}$$

Instead of optimization concerning w, Salimans and Kingma [125] optimize concerning g and v. The corresponding derivatives take the form:

$$\nabla_g L = \frac{\nabla_{\mathrm{w}} L\ .\mathrm{v}}{||\mathrm{v}||},\quad \nabla_{\mathrm{v}} L = \frac{g}{||\mathrm{v}||}\nabla_{\mathrm{w}} L - \frac{g\nabla_g L}{|||\mathrm{v}||^2}\mathrm{v}, \tag{21}$$

Where $\nabla_{\mathrm{w}} L$ is the gradient concerning the weights w. This equation shows the derivative for g.

$$\nabla_{\mathrm{v}} L = \frac{g}{||\mathrm{v}||} M_{\mathrm{w}} \nabla_{\mathrm{w}} L,\ \text{with } \mathrm{M_w} = \mathrm{I} - \frac{\mathrm{ww}'}{|||\mathrm{w}||^2} \tag{22}$$

Where $\nabla_{\mathrm{v}} L$ is the gradient concerning the weights v. Equation (22) shows the derivative for v. Note, $\mathrm{M_w}$ takes the form of a house-holder transformation on the normalized vector $\mathrm{w}/\|\mathrm{w}\|^2$.

3.5 Instance Normalization (IN)

The **instance normalization (IN)** approach [126]. normalizes each example's features instead of normalizing all the features at once across mini-batches. Quality of style transfer further improved with the use of IN. The performance of specific deep neural networks in GAN, like image generation applications, dramatically improves through IN due to batch-size limitations.

3.6 Group Normalization (GN)

Group normalization (GN) [127] is a simple substitute for the famous BN technique, which suffered from varying batch dimensions. Instead of normalizing each channel's GN features, they divide the channel and then normalize the features.

3.7 Batch-instance Normalization (BIN)

Batch-instance normalization technique (BIN) [128] proposed to normalize the redundant styles from images by binding the advantages of both batch normalization (BN) [122] and instance normalization (IN) [126]. The IN discards the irrelevant information in BIN, and the BN preserves the critical, relevant information. So, both normalization techniques are in collaboration with BIN for better results.

Let us suppose x $\in \mathbb{R}^{NxCxHxW}$ to be an input mini-batch and x_{nchw} denotes the element to a specific layer, where n represents mini-batch sample index, c represents index channel, and w and h specifies the spatial position.

At the first stage, the BN technique normalizes each example of the mini-batch through mean and variance:

$$\hat{x}_{nchw}^{(B)} = \frac{x_{nchw-}\ \mu_c^{(B)}}{\sqrt{\sigma_c^{2(B)} + \epsilon}}, \tag{23}$$

$$\hat{x}_{nchw}^{(B)} = \frac{x_{nchw-}\ \mu_c^{(B)}}{\sqrt{\sigma_c^{2(B)} + \epsilon}} \tag{24}$$

$$\sigma_{nc}^{2(I)} = \frac{1}{HW} \sum_H \sum_W \left(x_{nchw} - \mu_{nc}^{(I)}\right)^2, \tag{25}$$

Where $\hat{x}^{(B)} = \{\hat{x}_{nchw}^{(B)}\}$ is the result of BN, which discards irrelevant information.

At the second stage, IN technique normalizes each example through feature statistics of each instance:

$$\hat{x}_{nchw}^{(I)} = \frac{x_{nchw-}\ \mu_{nc}^{(I)}}{\sqrt{\sigma_{nc}^{2(I)} + \epsilon}}, \tag{26}$$

$$\mu_{nc}^{(I)} = \frac{1}{HW} \sum_H \sum_W (x_{nchw}), \tag{27}$$

$$\sigma_{nc}^{2(I)} = \frac{1}{HW} \sum_H \sum_W (x_{nchw} - \mu_{nc}^{(I)})^2, \tag{28}$$

Where $\hat{x}^{(I)} = \{\hat{x}_{nchw}^{(I)}\}$ is the result of IN, which preserves the critical, relevant information.

$$y = \left(\rho.\hat{\mathrm{x}}^{(B)} + (1-\rho)\,.\hat{\mathrm{x}}^{(I)}\right).\gamma + \beta, \tag{29}$$

BIN has to preserve important style attribute, and they do this with the introduction of learnable parameters, $\rho \in [0,\ 1]^C$:

$$\rho \leftarrow \mathrm{clip}_{[0,1]}\,(\rho - \eta\Delta\rho) \tag{30}$$

Where affine transformation parameters are represented by $\gamma,\ \beta \in \mathbb{R}^c$ and BIN output is y $\in$ R $^{N \times C \times\ H \times W}$.

Where $\Delta\rho$ shows the parameter update vector determined by the optimizer, η is the learning rate. Intuitively, ρ can be interpreted as a decider, i.e., the value of $\rho = 1$ if the style (BN preserves the style) is vital to the task, and the value of $\rho = 0$ if a style (the style is normalized through IN) is unnecessary.

3.8 Switchable Normalization (SN)

Switchable normalization (SN) [129] introduces how to tackle the different normalization techniques used for different convolutional layers within a **convolutional neural network (CNN)** [19]. The SN technique learns to select appropriate normalizations technique such as **batch normalization (BN)** Error! Reference source not found., **instance normalization (IN)** Error! Reference source not found., and **layer normalization (LN)** Error! Reference source not found. for each normalization layer of a CNN. Different experiments have shown better performance if each normalization layer uses its normalization operation.

3.9 Spectral Normalization (SN)

Spectral normalization (SN) [130] was proposed to stabilize GAN training. It is a weight normalization technique typically used on the discriminator where SN adjusts the weights of all the layers to 1 and always performs better. This essentially ensures that the discriminator is K-Lipchitz continuous. The EB-GAN [8] further proved that SN in the generator could also stop the acceleration of parameter magnitudes and avoid unusual gradients. The SN considerably reduces the computational cost and shows better training behavior because other hyper-parameters don't need to be tuned. The SN is defined as:

$$\bar{w}_{SN}\left(w\right) = \frac{W}{\sigma\left(W\right)} \tag{31}$$

Where $\bar{W}$ and $\sigma(W)$ corresponds to each layer for D and L_2 matrix normalization of W.

4 ANALYSIS OF GAN TRAINING OBSTACLES

This section provides an analysis to understand better the training issues' main reasoning with the appropriate remedies discussed in this survey. Table 3 presents a summary of the appropriate training technique for the appropriate scenario.

The literature review shows that GAN's [1] performance is severely damaged due to training shortcomings such as Nash equilibrium [143], which describes a particular state where both G and D do their best to have their best outcomes. GAN's optimization objective is to reach a Nash equilibrium stage where both the networks peaked concerning their performance. But practically, to reach this point is very difficult due to a lack of mutual communication between them. Both networks have to communicate with each other to enhance individual performance. Still, when both the networks didn't communicate with each other, or update their cost function independently without coordination, it is hard to achieve Nash equilibrium. So, the lack of mutual communication is the foremost hurdle to reach a Nash equilibrium point. Many methods, such as the W-GAN [9], **Two Time-scale Update Rule (TTUR)** [132], and **Least-Squares GAN (LS-GAN)** [142], are considered to reach the Nash equilibrium.

Internal covariate shift (ICS) [122] refers to constant adjustment in the distribution of inputs to the network layer during system training. The ICS slows down the training process, and the network requires a longer training time to converge to global minima. The main reasoning behind the ICS phenomena is the continuous change in network parameter values. The network becomes deeper due to constant changes to the network parameters values. **Batch Normalization (BN)** [122] and **Weight Normalization (WN)** [125] training techniques solve this problem through layer normalization strategy. **Mode collapse (MC)** [144, 145] describes how a generator generates samples with very little diversity. MC's main reasoning is that both G and D get more durable and more reliable with training iterations. But suppose one network, i.e., G or D, becomes far more influential than the other. In that case, the learning signal to the other network becomes ineffective, and the system fails to replicate the diversity in the training dataset. The

Table 3. A Summary of GANs Training Techniques' Pros and Cons

Model	Pros.	Cons.
Feature matching [123]	1. Stabilizes GAN training by giving the generator a new objective function. 2. Minimize the distance of intermediate features from multi-scale discriminators between real and fake images.	1. It has no fixed point where G exactly matches the distribution of training data.
Unrolled GAN [131]	1. Demonstrates that high-order gradient information can help in the training of GAN. 2. Increases diversity and coverage of the generated distribution by the generator.	1. The quality of the generated images is low. 2. Computational cost scales with the number of unrolling steps.
Minibatch discrimination [123]	1. Generate visually appealing samples very quickly. 2. Encourage the discriminator to consider multiple samples in combination instead of an individual sample.	1. There is no obvious way that minibatch discrimination could not penalize a collapsed conditional generator.
Two time-scale update rule [132]	1. Speed-up the GAN training process. 2. Balance the training process of the generator and discriminator networks. 3. Improves GAN learning ability.	1. The handling of separate learning parameters is a difficult task.
Self-attention mechanism [135]	1. Excellent performance on multi-class image generation. 2. Improves training dynamic by capturing the local and global dependencies of the target distribution.	1. Difficult to implement.
Hybrid model [133, 134]	1. Power of different models combined. 2. Handling complicated situations well.	1. Jointly training two models is a challenging task. 2. Computationally expensive.
Relativistic GAN [136]	1. A unified framework for IPM-based GANs models. 2. R-GANs can be generalized to any IPM-based GANs models.	1. Lack of mathematical implications of adding relativism to GANs. 2. Has not done a survey on which IPM-based GAN will achieve best.
One-sided label smoothing [137]	1. Reduce the vulnerability of GAN. 2. It is hindering the discriminator from becoming overconfident and squashing the generator's gradients.	1. Efficiency was dependent on the size of the training dataset.
Proper optimizer [139]	1. Improves the GAN learning rate. 2. Stabilizes GAN training.	1. No standard mechanism to decide which optimizer best fit for which task.
Batch normalization [122]	1. Improve the optimization of the model by parameterizing the model. 2. Creates a correlation among the data examples within a minibatch.	1. Efficiency was dependent on mini-batch size. 2. The model can overfit to the reference batch.
Spectral normalization [130]	1. Easy to implement and computationally light. 2. Stabilizes GAN training and improves the mode diversity of the model. 3. Spectral normalization is capable of being generalized to any GANs' variants.	1. The success of their model is related to the availability of a large database, so for a small amount of training data, the model's performance is not reasonable.
Add noise to Inputs [140, 141]	1. Make the discriminator less prone to over-fitting. 2. Improves GAN convergence power.	1. Introduces significant variance in the parameter estimation process with high computational cost. 2. The quality of the generated images is low.
Train with labels [2]	1. Provide more control over the required result. 2. Provide a general framework where the generator can generate images of distinct features.	1. Computationally expensive. 2. It required a labeled dataset.
Alternative loss functions [142]	1. Easy to implement. 2. Improves the model of diversity by stabilizes GAN training. 3. Generate better gradients to update its generator.	1. The quality of the generated images is low.

(Continued)

Table 3. Continued

Model	Pros.	Cons.
Gradient penalty [10]	1. Improves GAN convergence power. 2. Stabilizes GAN training. 3. Improves GAN training speed and sample quality.	1. The training loss of GAN gradually increases even while the validation loss drops.
Representative features [138]	1. Improve the visual quality of the generated samples. 2. Representative features are also used along with discriminative features to train the discriminator.	1. The representative features become less informative when approaching the second half of training because the AEs have limited performance for discrimination.
Cycle-consistency loss [94]	1. Matching pairs of images are no longer needed for training. 2. Minimize the reconstruction loss and stabilizes GAN training.	1. Prefer to learn one-to-one mappings.

other main reasoning is that the generator is not working according to its full capacity; the result is that it does not include all possible targeting modes. Different remedies have been proposed to reduce MC effects during training, such as WGAN [9], Mini-Batch discrimination [123], **Unrolled GAN (U-GAN)** [131], **Deep Regret Analytic GAN (DRA-GAN)** [146], Ada-GAN [147], **Mode Regularized GAN (MR-GAN)** [148], **Multi-Agent Diverse GAN (MAD-GAN)** [149] and try to strengthen the generator network to expand its capacity by restricting it from optimizing for a single rigid discriminator.

GAN is considered hard to train because of the **vanishing gradient** problem **(VG)** [144]. VG's main reasoning is the use of the sigmoid function in the standard GAN objective function for the discriminator network, and a powerful discriminator network can be the reasoning for the VG problem. The W-GAN [9], **Least Squares GAN (LS-GAN)** [142], **Loss Sensitive GAN (LS-GAN)** [150], **Relativistic GAN (R-GAN)** [136], **Spectral Normalization (SN-GAN)** [130], and **Batch Normalization (BN)** [122] techniques are designed to avert vanishing gradients problem for the generator network during GAN training process which improves the training stability and convergence ability of the system.

We can conclude from Table 3 that each training stabilization method has its pros and cons. Prior knowledge about the training technique's strengths and limitations may be crucial before applying it successfully to stabilize GAN's training process. This kind of in-depth knowledge improves their models' efficiency because the models' efficiency depends on stable training. It is also helpful when most training methods have a specific role in avoiding training-specific issues.

5 ANALYSIS OF GAN TRAINING TECHNIQUES

This section provides an in-depth analysis of GAN training techniques regarding their pros and cons to understand their strengths and limitations better because each stabilization technique is appropriate for a particular situation/ problem. Table 4 presents a summary of the pros and cons of the GAN training techniques.

The feature matching training technique of GAN [123] prevents the generator from over-fitting with the current discriminator by pressuring the generator to maximize the likelihood of the discriminator classifying the generated sample as real and to maximize the likelihood that the generated samples will match the required target distribution. After capturing real and fake samples' discriminatory features, feature matching GAN mitigates mode collapse in GAN training. It provides better results, even with a limited number of dataset labels. The **GAN unrolling training technique (U-GAN)** [131] adjusts the GAN training processes to solve the mode collapse problem

Table 4. A Summary of the Training Technique for the Appropriate Scenario

Training technique	Situation/Problem
Feature matching (FM) [123]	◇ Most appropriate for a situation where the generator may perform overtraining on the given discriminator. ◇ Most appropriate to avoid mode collapse problem.
Historical averaging (HA) [123]	◇ Most appropriate to avoid the situation where GANs have oscillating and cyclical behavior. ◇ Most appropriate to avoid mode collapse problem.
Unrolled GAN (UGAN) [131]	◇ Most appropriate for a situation where the training of the generator and discriminator needs to be adjusted to stabilize GANs. ◇ Most appropriate to avoid mode collapse problem.
Minibatch discrimination [123]	◇ Most appropriate for a situation where the discriminator needs to look at multiple examples in a mini-batch to stabilize GANs. ◇ Most appropriate to avoid mode collapse problem.
Time-scale update rule (TTUR) [132]	◇ Most appropriate for a situation where different learning rates required for generator and discriminator to stable GAN training. ◇ Most appropriate for the search of Nash-equilibrium stage (Saddle point).
Self-attention GAN (SA-GAN) [135]	◇ Most appropriate for a situation where the information from a broader feature space across image regions required for image generation task ◇ Most appropriate to avoid mode collapse problem.
Relativistic GAN (RGAN) [136]	◇ Unified framework for IPM-based GANs ◇ Most appropriate to avoid mode collapse and vanishing gradient problems.
One-sided label smoothing [137]	◇ Most appropriate to prevent the discriminator from becoming overconfident that would provide weak gradients for the generator. ◇ Most appropriate to avoid vanishing gradient problem.
Hybrid model [133, 134]	◇ Most appropriate for a situation where generative models have to deal with more complex and high-dimensional data distributions. ◇ Most appropriate to alleviate the mode collapse problem.
Proper optimizer (Adam) [139]	◇ Most appropriate for a situation where greater convergence speed required to stabilize GAN training. ◇ Most appropriate to avoid mode collapse problem.
Batch normalization (BN) [122]	◇ Most appropriate to greatly improve the optimization of neural networks through normalizing the activations of each layer, allowing each layer to learn more stable distribution of inputs. ◇ Most appropriate to avoid vanishing gradient and internal covariant-shift problems.
Spectral normalization (SN) [130]	◇ Most appropriate to generate images with more diversity and complexity than those generated with weight normalization. ◇ Most appropriate to avoid mode collapse and vanishing gradient problems.
Add noise to inputs [140, 141]	◇ Most appropriate to reduce the problem of non-overlapping support. ◇ Most appropriate to avoid vanishing gradient problem.
Train with labels [2]	◇ Most appropriate to increase the probability of correct class prediction. ◇ Most appropriate to avoid mode collapse problem.
Alternative loss functions [142]	◇ Most appropriate for a situation where penalizing the samples lying a long way to the decision boundary can generate more gradients when updating the generator, ◇ Most appropriate to avoid vanishing gradient problem.
Gradient penalty (WGAN-GP) [10]	◇ Most appropriate for a situation where discriminator network need to learn smoother decision boundaries for better convergence power, improved training speed, and better sample quality. ◇ Most appropriate to avoid mode collapse problem.
Representative features (RF) [138]	◇ Most appropriate for a situation where there is a need of hindering the fast discriminator update growth, thus achieving stable training. ◇ Most appropriate to avoid mode collapse problem.
Cycle-consistency loss [94]	◇ Most appropriate to prevent the generators from extreme hallucinations situation where paired training is not available. ◇ Most appropriate to avoid mode collapse problem.

by unrolling the discriminator network updates that improve the generator network's efficiency. U-GAN's generator network stops the discriminator network from reflecting on the last update; however, it updates its generations through the discrimination prospect.

GAN's mini-batch discrimination training technique [123] implies a good discriminator by allowing the discriminator to take numerous samples in a mini-batch instead of separately using a single data sample for discrimination purposes to prevent a generator from mode collapse. This way, the discriminator decides whether the batch of inputs is real or fake. The historical averaging training technique of GAN [123] prevents the mode collapse training obstacle by limiting the wobbling and cyclical behavior of GAN. The historical averaging GAN makes it feasible to generate more recognizable objects than the many other training techniques. The two time-scale update rule training technique of GAN [132] uses different learning rates for the generator and the discriminator searching for a saddle point (Nash equilibrium). They experimentally showed that the faster learning rate of the discriminator as compared to the generator learning rate is better for the stable training process of GAN. The **variational hybrid training method (VAE-GAN)** [133, 134] incorporates the advantages of auto-encoders and adversarial training to tackle a more complex and high-dimensional distribution of the data. The VAE-GAN hybrid approaches to avoid the mode collapse issue during model training.

The **self-attention training technique of GAN (SA-GAN)** [135] generates multi-class images by coordinating the fine details of every location with distant portions. The SA-GAN prevents the model from collapsing to stabilize the training process of GAN. The **relativistic (R-GAN) training technique** [136] makes the GAN training process stable and better compared to vanilla GAN by pushing the discriminator network towards 0.5 rather than towards 1. The R-GAN's discriminator predicts relative, rather than absolute, realness. The relativistic (R-GAN) based training approach alleviates the vanishing gradient problem. The one-sided label smoothing training technique of GAN [137] encourages the discriminator network to produce the probabilities between 0.1 (for generator generated samples) and 0.9 (for real samples) instead of 0 and 1. GAN's one-sided label smoothing training technique mitigates the issue of vanishing gradient in GAN training and provides better results even with a small number of dataset labels. The appropriate optimizer (Adam) [139] will increase the convergence rate necessary to stabilize GAN's training process. The Adam optimizer alleviates the mode collapse problem better than other optimizers.

The **batch normalization training technique of GAN (BN-GAN)** [122] can produce higher quality and diverse images by simply normalizing each layer's activations to stabilize the training process in deeper models without collapsing and reducing the internal covariant change between real and fake data. GAN's training with-labels [2] extra control over the outputs to dictate the type of data generated increases its data generation and data discrimination capacities. GAN's training with labels approaches alleviates the mode collapse issue. The **least-squares training technique of GAN (LS-GAN)** [142] prevents the vanishing gradient training obstacle in GAN training by simply replacing the sigmoid cross-entropy with the least-square because the sigmoid function in the discriminator is the leading cause of the vanishing gradient problem. However, the LS-GAN creates images that are not very clear because the images generated may change from good to low. The **gradient penalty (GP) training technique of GAN (WGAN-GP)** [10] enhances the training speed and quality of the sample by simply adding a gradient penalty (GP) term in place of the weight clipping. The term gradient penalty (GP) allows the discriminator to learn more about smoother decision boundaries and reduce the network mode to collapse.

The **spectral normalization** training technique of GAN **(SN-GAN)** [130] can generate more diverse and complex images by stabilizing the training process of the GAN's discriminator by adjusting the weights of all the layers to 1. The discriminator with normalized weight layers plays a significant role in stabilizing the training process of GAN. The SN-GAN's computational cost is low and doesn't require changing other hyper-parameters. This weight **normalization technique (N-GAN)** prevents mode collapse and vanishing gradient training obstacles. Adding instance noise training technique of GAN [140, 141] stabilizes the training process of GAN by making the discriminator network less susceptible to over-fitting, as it now has a broader range of training, which also helps in alleviating the vanishing gradient problem. The **representative features-based GAN (RF-GAN)** [138] breaks the trade-off between the sample diversity and the generated images' quality. The RF-GAN regularizes the discriminator network training to mitigate the mode collapse problem via pre-trained encoder features. The **cycle-consistency loss of GAN (Cycle-GAN)** [94] plays a significant role in restricting mode collapse and stabilizing the training process of GAN. The Cycle-GAN prevents mode collapse in the absence of paired examples using the combined loss of cycle-consistent training and adversarial training.

We can conclude from Table 4 that although many training stabilization methods have been introduced for several years to overcome GAN training instability issues and enhance the diversity of the generated samples with higher visual quality, most of these GANs training stabilization methods are limited to avoiding just one specific issue of training, and often have a short rigorous theoretical explanation. A stable training technique viable for different training obstacles that can be encountered when training GAN is missing. Thus, the development of a better and universal training algorithm for this scenario still requires future research.

REFERENCES

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio§. 2014. Generative adversarial nets. In *Neural Information Processing Systems* 2672–2680.

[2] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv: 1411.1784*, 2014.

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

[4] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *Neural Information Processing Systems* 1486–1494.

[5] P. Burt and E. Adelson. 1983. The laplacian pyramid as a compact image code. In *IEEE Transactions on Communications* 31, 4 532–540.

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever‡, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Neural Information Processing Systems* 2172–2180.

[7] Spurr E. Aksan and O. Hilliges. 2017. Guiding infoGAN with semi-supervision. *arXiv preprint arXiv: 1707.04487*, 2017.

[8] Junbo Zhao, Michael Mathieu, and Yann LeCun. 2016. Energy-based generative adversarial network. *arXiv preprint arXiv: 1609.03126*, 2016.

[9] Martin Arjovsky, Soumith Chintala, and Leon Bottou. 2017. Wasserstein generative adversarial networks. In *International Conference on Machine Learning* 214–223.

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of Wasserstein GANS. In *Neural Information Processing Systems* 5767–5777.

[11] C. Villani. 2009. Optimal transport: Old and new (grundlehren der mathematischen wissenschaften), 2009.

[12] David Berthelot, Thomas Schumm, and Luke Met. 2017. Began: Boundary equilibrium generative adversarial networks, arXiv preprint arXiv: 1703.10717, 2017.

[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive growing of GANs for improved quality, stability, and variation, *arXiv preprint arXiv: 1710.10196*, 2018.

[14] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[15] J. Donahue and K. Simonyan. 2019. Large scale adversarial representation learning. *arXiv preprint arXiv: 1907.02544*, 2019.

[16] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition* 4401–4410.

[17] X. Huang and S. J. Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision*. 2017.

[18] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. 2019. Analyzing and improving the image quality of styleGAN. *arXiv preprint arXiv: 1912.04958*, 2019.
[19] Alex Krizhevsky, I. Sutskever, and G. E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.
[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition* 1125–1134.
[21] Yuchen Tian. 2017. Zi2zi: Master Chinese calligraphy with conditional adversarial networks, 2017.
[22] Bo Chang, Qiong Zhang, Shenyi Pan, and Lili Meng. 2018. Generating handwritten Chinese characters using cycle-GAN. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.
[23] Xianming Lin, Jie Li, Hualin Zeng, and Rongrong Ji. 2018. Font generation based on least-squares conditional generative adversarial nets. In *Multimedia Tools and Applications*, 2018.
[24] Hideaki Hayashia, Kohtaro Abea, and Seiichi Uchidaa. 2019. GlyphGAN: Style-consistent font generation based on generative adversarial networks. *arXiv preprint arXiv: 1905.12502*, 2019.
[25] Jin Liu, Chenkai Gu, Jin Wang, Geumran Youn, and Jeong-Uk Kim. 2019. Multi-scale multi-class conditional generative adversarial network for handwritten character generation. In *Journal of Supercomputing* 4, 1922–1940.
[26] Bo Ji and Tianyi Chen. 2019. Generative adversarial network for handwritten text. *arXiv preprint arXiv: 1907.11845*, 2019.
[27] Md. Fahim Sikder. 2019. Bangla handwritten digit recognition and generation. In *Proceedings of International Joint Conference on Computational Intelligence* 547–556.
[28] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. 2017. Towards the automatic anime characters creation with generative adversarial networks. *arXiv preprint arXiv: 1708.05509*, 2017.
[29] Koichi Hamada, Kentaro Tachibana, Tianqi Li, Hiroto Honda, and Yusuke Uchida. 2018. Full-body high-resolution anime generation with progressive structure-conditional generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
[30] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. 2017. GpGAN: Towards realistic high-resolution image blending. *arXiv preprint arXiv: 1703.07195*, 2017.
[31] Bor-Chun Chen and Andrew Kae. 2019. Toward realistic image compositing with adversarial learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* 8407–8416.
[32] B. Dolhansky and C. C. Ferrer. 2018. Eye in-painting with exemplar generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition* 7902–7911.
[33] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 5505–5514.
[34] U. Demir and G. Unal. 2018. Patch-based image in-painting with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[35] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. 2017. Face aging with conditional generative adversarial networks. *arXiv preprint arXiv: 1702.01983*, 2017.
[36] Zhifei Zhang, Yang Song, and Hairong Qi. 2017. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[37] Zongwei Wang, Xu. Tang, Weixin Luo, and Shenghua Gao. 2018. Face aging with identity-preserved conditional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
[38] Peipei Li, Yibo Hu, Ran He, and Zhenan Sun. 2018. Global and local consistent wavelet-domain age synthesis. *arXiv preprint arXiv: 1809.07764*, 2018.
[39] Yunfan Liu, Qi Li1, and Zhenan Sun. 2019. Attribute aware face aging with wavelet-based generative adversarial networks. arXiv *preprint arXiv: 1809.06647*, 2019.
[40] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2018. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[41] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. 2016. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv: 1612.03242*, 2016.
[42] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier GANs. In *International Conference on Machine Learning*, 2017.
[43] Ayushman Dash, John Gamboa, and Sheraz Ahmed. 2017. Tac-GAN-text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv: 1703.06412*, 2017.
[44] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. 2017. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision* 5706–5714.

[45] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. 2018. Pose guided person image generation. *arXiv preprint arXiv: 1705.09368*, 2018.
[46] Aliaksandr Siarohin, Stephane Lathuiliere, Enver Sangineto, and Nicu Sebe. 2019. Appearance and pose conditioned human image generation using deformable GANs. *arXiv preprint arXiv: 1905.00007*, 2019.
[47] Denis Volkhonski, Ivan Nazarov, Boris Borisenko, and Evgeny Burnaev. 2017. Steganographic generative adversarial networks. *arXiv preprint arXiv: 1703.05502*, 2017.
[48] Haichao Shia, Jing Dongc, Wei Wangc, Yinlong Qianc, and Xiaoyu Zhanga. 2017. SsGAN: Secure steganography based on generative adversarial networks. *arXiv preprint arXiv: 1707.01613*, 2017.
[49] Kevin A. Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. 2019. Steganogan: High capacity image steganography with GANs. *arXiv preprint arXiv: 1901.03892*, 2019.
[50] Jun-Yan Zhu, Philipp Kr¨ahenb¨uhl, Eli Shechtman, and Alexei A. Efros1. 2016. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613, 2016.
[51] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. 2018. Text-adaptive generative adversarial networks: Manipulating images with natural language. In *Proceedings of International Conference on Neural Information Processing*, 2018.
[52] Andrew Brock, Theodore Lim, J. M. Ritchie, and Nick Weston. 2016. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv: 1609.07093*, 2016.
[53] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. 2019. AttGAN: Facial attribute editing by only changing what you want. In *IEEE Transactions on Image Processing*, 2019.
[54] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. 2019. GAN dissection: Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations*, 2019.
[55] Junting Pana, Cristian Canton-Ferrerb, Kevin McGuinnessc, Noel E. O'Connorc, Jordi Torresd, Elisa Sayrola, and Xavier Giro-i-Nieto. 2017. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017.
[56] Chao Zhang, Fei Yang, Guoping Qiu, and Qian Zhang. 2019. Salient object detection with capsule-based conditional generative adversarial network. In *IEEE International Conference on Image Processing* 2019.
[57] Prerana Mukherjee, Manoj Sharma, Megh Makwana, Ajay Pratap Singh, Avinash Upadhyay, Akkshita Trivedi, Brejesh Lall, and Santanu Chaudhury. 2019. Dsal-GAN: Denoising based saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1904*.01215, 2019.
[58] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. 2018. Segan: Segmenting and generating the invisible. In *IEEE Conference on Computer Vision and Pattern Recognition* 6144–6153.
[59] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. 2017. Perceptual generative adversarial networks for small object detection. In *IEEE Conference on Computer Vision and Pattern Recognition* 1222–1230.
[60] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. 2018. Sod-mtGAN: Small object detection via a multi-task generative adversarial network. In *Proceedings of the European Conference on Computer Vision* 206–221.
[61] Charan D. Prakash and Lina J. Karam. 2019. It GANdo better: GAN based detection of objects on images with varying quality. *arXiv preprint arXiv: 1912.01707*, 2019.
[62] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. 2016. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, 2016.
[63] Matheus Gadelha, Subhransu Maji, and Rui Wang. 2016. 3d shape induction from 2d views of multiple objects. *arXiv preprint arXiv: 1612.05872*, 2016.
[64] Cihan Ongun and Alptekin Temizel. 2018. Paired 3d model generation with conditional generative adversarial networks. In *Proceeding of the 15th European Conference on Computer Vision (ECCV)*, 2018.
[65] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 2019. 3d point cloud generative adversarial network-based on tree-structured graph convolutions. In *Proceedings of the Conference on Computer Vision*, 2019.
[66] Pranav Gandhi, Adarsh Shaw, Emil Eji, and Steffina Muthukumar. 2019. Generating 3d models using 3d generative adversarial network. In *International Research Journal of Engineering and Technology*
[67] Yuan Xue, Tao Xu, Han Zhang, L. Rodney Long, and Xiaolei Huang. 2018. SeGAN: Adversarial network with multi-scale $l1$- loss for medical image segmentation. In *Neuro-informatics* 16, 383–392.
[68] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. 2017. Generating multi-label discrete patient records using generative adversarial networks. *arXiv preprint arXiv: 1703.06490*, 2017.
[69] Mostapha Benhenda. 2017. ChemGAN challenge for drug discovery: Can AI reproduce natural chemical diversity?. *arXiv preprint arXiv: 1708.08227*, 2017.
[70] Anvita Gupta and James Zou. 2018. Feedback GAN (fbGAN) for DNA: A novel feedback-loop architecture for optimizing protein functions. *arXiv preprint arXiv:1804.01694*, 2018.

[71] Ghislain St-Yves and Thomas Naselaris. 2018. Generative adversarial networks conditioned on brain activity reconstruct seen images. In *Conference on Systems, Man, and Cybernetics* 1054–1061.
[72] Jyh-Jing Hwang, Sergei Azernikov, Alexei A. Efros, and Stella X. Yu. 2018. Learning beyond human expertise with generative models for dental restorations. *arXiv preprint arXiv: 1804.00064*.
[73] Bing Tian, Yong Zhang, Xinhuan Chen, Chunxiao Xing, and Chao Li. 2019. Drgan: A GAN based framework for doctor recommendation in Chinese on-line qa communities. In *International Conference on Database Systems for Advanced Applications* 444–447.
[74] Karim Armanious, Chenming Yang, Marc Fischer, Thomas K¨ustner, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang. 2020. Medgan: Medical image translation using GAN. In *Computerized Medical Imaging and Graphics*, 2020.
[75] Tingting Li, Ruihe Qian, Chao Dong, Si Liu, Qiong Yan, Wenwu. Zhu, and Liang Lin. 2018. BeautyGAN: Instance-level facial makeup transfer with the deep generative adversarial network. In *ACM Multimedia Conference on Multimedia Conference* 645–653.
[76] Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. 2018. PairedCycleGAN: Asymmetric style transfer for applying and removing makeup. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 40–48.
[77] Honglun Zhang, Jidong Tian, Wenqing Chen, Hao He, and Yaohui Jin. 2019. Disentangled makeup transfer with generative adversarial network. *arXiv preprint arXiv: 1907.01144*, 2019.
[78] Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. 2018. Style aggregated network for facial landmark detection. *arXiv preprint arXiv: 1803.04108*.
[79] Xin Yang, Yuezun Li, Honggang Qi, and Siwei Lyu. 2019. Exposing GAN synthesized faces using landmark locations. *arXiv preprint arXiv: 1904.00167*.
[80] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, and Andrew Cunningham. 2017. Photorealistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition* 4681–4690.
[81] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. 2018. EsrGAN: Enhanced super-resolution generative adversarial networks. In *European Conference on Computer Vision* 63–79.
[82] Jingwei Guan, Cheng Pan, Songnan Li, and Dahai Yu. 2019. Srdgan: Learning the noise prior for super-resolution with dual generative adversarial networks. *arXiv preprint arXiv: 1903.11821*, 2019.
[83] Zihan Ding, Xiao-Yang Liu, Miao Yin, W. Liu, and Linghe Kong. 2019. TGAN: Deep tensor generative adversarial nets for large image generation. *arXiv preprint arXiv: 1901.09953*, 2019.
[84] Chuan Li and Michael Wand. 2016. Precomputed real-time texture synthesis with Markovian generative adversarial networks. In *European Conference on Computer Vision* 702–716.
[85] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. Texture synthesis using CNN. In *Advances in Neural Information Processing Systems*, 2015.
[86] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. 2016. Texture synthesis with spatial generative adversarial networks. *arXiv preprint arXiv: 1611.08207*, 2016.
[87] Urs Bergmann, Nikolay Jetchev, and Roland Vollgraf. 2017. Learning texture manifolds with the periodic spatial GAN. In *Proceedings of the International Conference on Machine Learning* 469–477.
[88] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. 2017. TextureGAN: Controlling deep image synthesis with texture patches. *arXiv preprint arXiv: 1706.02823*, 2017.
[89] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. 2017. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition,* 2017.
[90] Yifan Liu, Zengchang Qin, Zhenbo Luo, and Hua Wang. 2017. Autopainter: Cartoon image generation from sketch by using the conditional generative adversarial networks. *arXiv preprint arXiv 1705.01908*, 2017.
[91] Wengling Chen and James Hays. 2018. Sketchygan: Towards diverse and realistic sketch to image synthesis. *arXiv preprint arXiv: 1801.02753*, 2018.
[92] Jun Yu, Xingxin Xu, Fei Gao, Shengjie Shi, Meng Wang, Dacheng Tao, and Qingming Huang. 2020. Towards realistic face photo-sketch synthesis via composition aided GAN. *arXiv preprint arXiv: 1712.00899*, 2020.
[93] Chaoyue Wang, Chang Xu, Chaohui Wang, and Dacheng Tao. 2018. Perceptual adversarial networks for image-to-image transformation. In *IEEE Transactions on Image Processing* 27, 8 (2018), 4066–4079.
[94] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision* 2017.
[95] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. 2017. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv: 1703.05192*, 2017.
[96] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. DualGAN: Unsupervised dual learning for image-to-image translation. *arXiv preprint arXiv: 1704.02510*.

[97] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition* 8789–8797.

[98] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*. 2017.

[99] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision (ECCV)*, 172–189.

[100] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2018. Diverse image-to-image translation via disentangled representations. In *European Conference on Computer Vision* 35–51.

[101] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. 2019. Drit++: Diverse image-to-image translation via disentangled representations. *arXiv preprint arXiv: 1905.01270*, 2019.

[102] Luan Tran, Xi Yin, and Xiaoming Liu. 2017. Disentangled representation learning GAN for pose-invariant face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* 1415–1424.

[103] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. 2017. Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis. In *International Conference on Computer Vision* 2439–2448.

[104] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. 2017. Towards large-pose face frontalization in the wild. In *IEEE International Conference on Computer Vision (ICCV)*. 2017.

[105] Volker Blan and Thomas Vetter. 1999. A morphable model for the synthesize of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques,* 187–194.

[106] Weiwei Zhuang, Liang Chen, Chaoqun Hong, Yuxin Liang, and Keshou Wu. 2019. FtGAN: Face transformation with key points alignment for pose-invariant face recognition. In *Electronics (Basel)* 8, 7 (2019), 807.

[107] Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. *arXiv preprint arXiv: 1705.11001*, 2017.

[108] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. 2017. Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks. *arXiv preprint arXiv: 1704.00849*, 2017.

[109] Olof Mogren. 2016. C-rnn-GAN: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv: 1611.09904*, 2016.

[110] Sepp Hochreiter and J· urgen Schmidhuber. 1997. Long short-term memory. In *Neural Computation*, 1997.

[111] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *AAAI Conference on Artificial Intelligence* 2852–2858.

[112] Gabriel Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. 2017. Objective reinforced generative adversarial networks (orGANs) for sequence generation models. *arXiv preprint arXiv: 1705.10843*, 2017.

[113] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems* 613–621.

[114] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. MoCoGAN: Decomposing motion and content for video generation. In *IEEE Conference on Computer Vision and Pattern Recognition* 1526–1535.

[115] Emily Denton and Vighnesh Birodkar. 2017. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems* 4414–4423.

[116] Aidan Clark, Karen Simonyan, and Jeff Donahue. 2019. Adversarial video generation on complex datasets. *arXiv preprint arXiv: 1907.06571*, 2019.

[117] Alexander. Y. Sun. 2018. Discovering state-parameter mappings in subsurface models using generative adversarial networks. In *Geophysical Research Letters* 45, 20 (2018), 11–137

[118] Zhi Zhonga, Alexander. Y. Sun, and Xinming Wu. 2020. Inversion of time-lapse seismic reservoir monitoring data using cycleGAN: A deep learning-based approach for estimating 3 dynamic reservoir property changes. In *Journal of Geophysical Research: Solid Earth* 125, 3, 2020.

[119] Mario Rüttgers, Sangseung Lee, Soohwan Jeon, and Donghyun You. 2019. Prediction of a typhoon track using a generative adversarial network and satellite images. In *Scientific Reports* 9, 6057*,* 2019.

[120] Kun Qian, Abduallah Mohamed, and Christian Claudel. 2019. Physics informed data-driven model for flood prediction: Application of deep learning in the prediction of urban flood development. *arXiv preprint arXiv: 1908.10312*, 2019.

[121] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems* 3856–3866.

[122] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv: 1502.03167*, 2015.

[123] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *Neural Information Processing Systems* 2234–2242.

[124] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv preprint arXiv: 1607.06450*, 2016.

[125] Tim Salimans and Diederik P. Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *arXiv preprint arXiv: 1602.07868*, 2016.

[126] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv: 1607.08022*, 2016.

[127] Yuxin Wu and Kaiming He. 2018. Group normalization. *arXiv preprint arXiv: 1803.08494*, 2018.

[128] Hyeonseob Nam and Hyo-Eun Kim. 2018. Batch instance normalization for adaptively style invariant neural networks. *arXiv preprint arXiv: 1805.07925*, 2018.

[129] Ping Luo, Jiamin Ren Zhanglin Peng, Ruimao Zhang, and Jingyu Li. 2019. Differentiable learning to normalizes via switchable normalization. In *International Conference on Learning Representations (ICLR)*, 2019.

[130] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[131] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. 2016. Unrolled generative adversarial networks. *arXiv preprint arXiv: 1611.02163*, 2016.

[132] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GAN trained by a two time-scale update rule converge to local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

[133] Anders Boesen Lindbo Larse, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. 2015. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv: 1512.09300,* 2015.

[134] Jin-Young Kim, Seok-Jun Bu, and Sung-Bae Cho. 2018. Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. In *Information Sciences,* 460, 83-102.

[135] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2018. Self-attention generative adversarial networks. *arXiv preprint arXiv: 1805.08318*, 2018.

[136] Jolicoeur-Martineau. 2019. The relativistic discriminator: A key element missing from standard GAN. In *International Conference on Learning Representation*, 2019.

[137] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition* 2818–2826.

[138] Duhyeon Bang and Hyunjung Shim. 2018. Improved training of generative adversarial networks using representative features. In *International Conference on Machine Learning*, 2018.

[139] Diederik P. Kingma, and Jimmy Lei Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv: 1412.6980*, 2014.

[140] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. 2017. Amortised map inference for image super-resolution. In *International Conference on Learning Representations*, 2017.

[141] Martin Arjovsky and Leon Bottou. 2017. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.

[142] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision*, 2017.

[143] Lillian J. Ratliff, Samuel A. Burden, and S. Shankar Sastry. 2013. Characterization and computation of local nash equilibria in continuous games. In *Annual Allerton Conference on Communication, Control and Computing* 917–924.

[144] Ian Goodfellow. 2016. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv: 1701.00160*, 2016.

[145] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. 2017. Generalization and equilibrium in generative adversarial nets (GANs). In *International Conference on Machine Learning*, 2017.

[146] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. 2017. How to train your draGAN. *arXiv preprint arXiv: 1705.07215,* 2017.

[147] Ilya Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Sch¨olkopf. 2017. AdaGAN: Boosting generative models. In *Advances in Neural Information Processing Systems* 5424–5433.

[148] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. 2016. Mode regularized generative adversarial networks. *arXiv preprint arXiv: 1612.02136*, 2016.

[149] Arnab Ghosh, Viveka Kulharia, Vinay Namboodiri, Philip H. S. Torr, and Puneet K. Dokania. 2017. Multi-agent diverse generative adversarial networks. *arXiv preprint arXiv: 1704.02906*, 2017.

[150] G.-J. Qi. 2019. Loss-sensitive generative adversarial networks on Lipschitz densities. *International Journal of Computer Vision* 1–23.