A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications

Jie Gui, *Senior Member, IEEE*, Zhenan Sun, *Senior Member, IEEE*, Yonggang Wen, *Fellow, IEEE*, Dacheng Tao, *Fellow, IEEE*, and Jieping Ye, *Fellow, IEEE*

Abstract—Generative adversarial networks (GANs) have recently become a hot research topic; however, they have been studied since 2014, and a large number of algorithms have been proposed. Nevertheless, few comprehensive studies explain the connections among different GAN variants and how they have evolved. In this paper, we attempt to provide a review of the various GAN methods from the perspectives of algorithms, theory, and applications. First, the motivations, mathematical representations, and structures of most GAN algorithms are introduced in detail, and we compare their commonalities and differences. Second, theoretical issues related to GANs are investigated. Finally, typical applications of GANs in image processing and computer vision, natural language processing, music, speech and audio, the medical field, and data science are discussed.

Index Terms—Deep learning, generative adversarial networks, algorithm, theory, applications

1 Introduction

Generative adversarial networks (GANs) have recently become a hot research topic. Yann LeCun, a legend in deep learning, said in a Quora post "GANs are the most interesting idea in the last 10 years in machine learning." According to Google Scholar, a large number of papers related to GANs already exist. For example, approximately 28,500 papers related to GANs were published in 2020, constituting approximately 78 papers every day or more than three per hour.

GANs consist of two models: a generator and a discriminator. These two models are typically implemented using neural networks but could be implemented using any form of differentiable system that maps data from one space to another. The generator tries to capture the distribution of true examples and generate new data examples. The discriminator is usually a binary classifier used to discriminate generated examples from true examples as accurately as possible. The optimization of GANs is a minimax optimization problem. The optimization terminates at a saddle point that forms a minimum with respect to the generator and a maximum with respect to the discriminator. That is, the GAN optimization goal is to reach Nash equilibrium [1]. At that point, the generator can be considered to have accurately captured the distribution of real examples.

Some previous works adopted the concept of making two neural networks compete with each other. The most relevant works are adversarial curiosity [2], [3], [4] and predictability minimization [5]. The connections among adversarial curiosity, predictability minimization, and GANs can be found in [6], [7].

The popularity and importance of GANs have led to several previous reviews. The difference between this study and previous works is summarized below.

1) *GANs for specific applications*: Some surveys have targeted the use of GANs for specific applications, such as image synthesis and editing [8], text-to-image synthesis [9], and audio enhancement and synthesis [10].
2) *General surveys*: The earliest relevant review was probably the paper by Wang *et al.* [11], which introduced the progress in GANs before 2017. Others [12], [13] mainly covered the progress in GANs prior to 2018. The authors of [14] introduced architecture variants and loss variants of GANs related only to computer vision. Other related works can be found in [15], [16].

- *Jie Gui is with the School of Cyber Science and Engineering, Southeast University, Nanjing, Jiangsu 211100, China, and also with Purple Mountain Laboratories, Nanjing 210000, China. He was previously with the Department of Computational Medicine and Bioinformatics, University of Michigan, USA. E-mail: guijie@seu.edu.cn.*
- *Zhenan Sun is with the Center for Research on Intelligent Perception and Computing, Chinese Academy of Sciences, Beijing 100190, China. E-mail: znsun@nlpr.ia.ac.cn.*
- *Yonggang Wen is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798. E-mail: ygwen@ntu.edu.sg.*
- *Dacheng Tao is with JD Explore Academy, China, and also with the School of Computer Science in the University of Sydney, Australia. E-mail: dacheng.tao@gmail.com.*
- *Jieping Ye is with Beike, Beijing 100085, China, and also with The University of Michigan, Ann Arbor, MI 48109 USA. E-mail: jieping@gmail.com.*

Manuscript received 22 July 2021; revised 29 Oct. 2021; accepted 21 Nov. 2021. Date of publication 0 . 0000; date of current version 0 . 0000.
This work was supported in part by the National Science Foundation of China under Grants 62172090 and 62172089, in part by the National Key R&D Project of China under Grant 2021QY2102, in part by CAAI-Huawei MindSpore Open Fund, in part by Alibaba Group through Alibaba Innovative Research Program, in part by the Grant of the Singapore National Research Foundation (NRF) under Sustainable Tropical Data Center Testbed (STDCT) Project, and in part by Jiangsu Provincial Double-Innovation Doctor Program under Grant JSSCBS20210075.
(Corresponding author: Zhenan Sun.)
Recommended for acceptance by L. Chen.
Digital Object Identifier no. 10.1109/TKDE.2021.3130191

TABLE 1
Overview of GAN Algorithms Discussed in Section 3

GANs Representative variants		InfoGAN [17], cGANs [18], CycleGAN [19], f-GAN [20], WGAN [21], WGAN-GP [22], LSGAN [23]
GANs training	Objective function	LSGANs [24], [25], hinge loss based GANs [26]–[28], MDGAN [29], unrolled GAN [30], SN-GANs [26], RGANs [31]
	Skills	ImprovedGANs [32], AC-GAN [33]
	Structure	LAPGAN [34], DCGANs [35], PGGAN [36], StackedGAN [37], SAGAN [38], BigGANs [39], StyleGAN [40], hybrids of autoencoders and GANs (EBGAN [41], BEGAN [42], BiGAN [43]/ALI [44], AGE [45]), multi-discriminator learning (D2GAN [46], GMAN [47]), multi-generator learning (MGAN [48], MAD-GAN [49]), multi-GAN learning (CoGAN [50])

TABLE 2
Applications of GAN Algorithms Discussed in Section 5

Field	Subfield	Method
Image processing and computer vision	Super-resolution	SRGAN [51], ESRGAN [52], Cycle-in-Cycle GANs [53], SRDGAN [54], TGAN [55]
	Image synthesis and manipulation	DR-GAN [56], TP-GAN [57], PG^2 [58], PSGAN [59], APDrawingGAN [60], IGAN [61], introspective adversarial networks [62], GauGAN [63]
	Texture synthesis	MGAN [64], SGAN [65], PSGAN [66]
	Object detection	Segan [67], perceptual GAN [68], MTGAN [69]
	Video	VGAN [70], DRNET [71], Pose-GAN [72], video2video [73], MoCoGan [74]
Sequential data	Natural language processing (NLP)	RankGAN [75], IRGAN [76], [77], TAC-GAN [78]
	Music	RNN-GAN (C-RNN-GAN) [79], ORGAN [80], SeqGAN [81]

To the best of our knowledge, this paper is the first to provide a comprehensive survey of GANs from algorithm, theory, and application perspectives that covers recent progress. Furthermore, our paper focuses on applications related not only to image processing and computer vision but also to sequential data such as natural language processing and to related areas such as the medical field.

The remainder of this paper is organized as follows. The related works are discussed in Section 2. Sections 3–5 introduce GANs from the algorithm, theory, and application perspectives. Tables 1 and 2 list the main GAN algorithms and application fields, which are discussed in Sections 3 and 5, respectively. Finally, Section 6 concludes the survey.

2 Related Work

GANs belong to a class of generative algorithms. Generative algorithms and discriminative algorithms are two categories of machine learning algorithms. Approaches that explicitly or implicitly model the distributions of inputs as well as outputs are known as generative models [82]. Generative algorithms have become increasingly popular and important due to their wide practical applications.

2.1 Generative Algorithms

Generative algorithms can be classified into two classes: explicit density models and implicit density models.

2.1.1 Explicit Density Models

An explicit density model defines a probability density function $p_{model}(x;\theta)$ and utilizes true data to fit the parameters θ. After training, new examples are produced utilizing the trained model or distribution. The explicit density models include maximum likelihood estimation (MLE), approximate inference [83], [84], and the Markov chain method [85], [86], [87]. These explicit density models use an explicit distribution and have limitations. For instance, MLE is conducted on true data, and its parameters are directly updated based on the true data, which leads to an overly smooth generative model. The generative model learned by approximate inference only approaches the lower bound of the objective function rather than directly solving the objective function because of difficulties involved in solving the objective function. The Markov chain algorithm can be used to train generative models, but it is computationally expensive. Furthermore, explicit density models have a computational tractability problem because they may fail to reflect the complexity of the true data distribution and learn the high-dimensional data distributions [88].

2.1.2 Implicit Density Models

An implicit density model does not directly estimate or fit the data distribution; instead, it produces data instances from the distribution without an explicit hypothesis [89] and utilizes the produced examples to modify the model. Prior to GANs, the implicit density model generally needs to be trained utilizing either ancestral sampling [90] or Markov chain-based sampling, which is inefficient and limits their practical applications. GANs belong to the directed implicit density model category. A detailed summary and relevant papers can be found in [91].

2.1.3 Comparison of GANs and Other Generative Algorithms

GANs were proposed to overcome the disadvantages of other generative algorithms. The basic idea behind adversarial learning is that the generator tries to create examples that are as realistic as possible to deceive the discriminator, while the discriminator tries to distinguish the generated fake examples from true examples. Both the generator and discriminator are improved through adversarial learning. This adversarial process gives GANs notable advantages over other generative algorithms. The specific advantages of GANs over other generative algorithms are as follows.

1) GANs can parallelize generation across a single large image, which is difficult for other generative algorithms such as the pixel convolutional neural network (PixelCNN) [92] and fully visible belief networks (FVBNs) [93], [94].
2) The generator design has few restrictions.
3) GANs are subjectively thought to produce better examples than those produced by other methods.

Refer to [91] for more detailed discussions about these comparisons.

2.2 Adversarial Idea

The adversarial idea has been successfully applied in many areas, including machine learning, artificial intelligence, computer vision and natural language processing. The 2016 defeat of the world's top human Go player by the AlphaGo model [95] engaged public interest in artificial intelligence. The intermediate version of AlphaGo utilizes two networks that compete with each other.

Adversarial examples [96], [97], [98], [99], [100], [101], [102], [103], [104], [105] also involve the adversarial idea. Adversarial examples are examples that differ substantially from real examples but are classified into a real category with high confidence or examples that differ only slightly from the real examples but are misclassified. This has recently become a very hot research topic [100], [101]. To prevent adversarial attacks [106], [107], [108], [109] utilized GANs to conduct the correct defense.

Adversarial machine learning [110] is a minimax problem in which a defender, who builds the classifier that we want to work correctly, searches over the parameter space to find the parameters that reduce the cost of the classifier as much as possible. Simultaneously, the attacker searches over the model inputs to maximize the cost.

Adversarial ideas can be found in adversarial networks, adversarial machine learning, and adversarial examples. However, they have different objectives.

3 Algorithms

In this section, we first introduce the original GANs followed by their representative variants and training.

3.1 GANs

The GAN framework is straightforward to implement when the models are both neural networks. To learn the generator distribution p_g over data x, a prior on input noise variables is defined as $p_z(z)$ [6], where z is the noise variable. Then, the generator represents a mapping from noise space to data space as $G(z, \theta_g)$, where G is a differentiable function represented by a neural network with parameters θ_g. The other neural network, $D(x, \theta_d)$, is also defined with parameters θ_d, but the output of $D(x)$ is a single scalar. $D(x)$ denotes the probability that x comes from the data rather than from the generator G. The discriminator D is trained to maximize the probability of assigning a correct label to both real training data and fake examples generated by the generator G. Simultaneously, G is trained to minimize $\log(1 - D(G(z)))$.

3.1.1 *Objective Function*

Different objective functions can be used in GANs.

3.1.1.1 Original minimax game: The objective function of GANs [6] is

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \tag{1}$$

where $\log D(x)$ is the cross-entropy between $[1 \quad 0]^T$ and $[D(x) \quad 1 - D(x)]^T$. Similarly, $\log(1 - D(G(z)))$ is the cross-entropy between $[0 \quad 1]^T$ and $[D(G(z)) \quad 1 - D(G(z))]^T$. For a fixed G, the optimal discriminator D [6] is given by

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}. \tag{2}$$

The minimax game in (1) can be reformulated as

$$\begin{aligned} C(G) = & \max_D V(D,G) = E_{x\sim p_{data}}\left[\log D_G^*(x)\right] \\ & + E_{z\sim p_z}\left[\log\left(1 - D_G^*(G(z))\right)\right] \\ & = E_{x\sim p_{data}}\left[\log D_G^*(x)\right] + E_{x\sim p_g}\left[\log\left(1 - D_G^*(x)\right)\right] \\ & = E_{x\sim p_{data}}\left[\log \frac{p_{data}(x)}{\frac{1}{2}\left(p_{data}(x) + p_g(x)\right)}\right] \\ & + \ E_{x\sim p_g}\left[\frac{p_g(x)}{\frac{1}{2}\left(p_{data}(x) + p_g(x)\right)}\right] - 2\log 2. \end{aligned} \tag{3}$$

The Kullback–Leibler (KL) divergence and the Jensen-Shannon (JS) divergence between two probabilistic distributions $p(x)$ and $q(x)$ are defined as follows:

$$KL(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx, \tag{4}$$

$$JS(p\|q) = \frac{1}{2} KL\left(p \middle\| \frac{p+q}{2}\right) + \frac{1}{2} KL\left(q \middle\| \frac{p+q}{2}\right). \tag{5}$$

Therefore, (3) is equal to

$$\begin{aligned} C(G) &= KL\left(p_{data} \middle\| \frac{p_{data} + p_g}{2}\right) + KL\left(p_g \middle\| \frac{p_{data} + p_g}{2}\right) - 2\log 2 \\ &= 2JS(p_{data} \| p_g) - 2\log 2. \end{aligned} \tag{6}$$

Thus, the objective function of GANs is related to the JS divergence.

3.1.1.2 Non-saturating game: In some cases, Eq. (1) may not provide a sufficient gradient for G to learn well. Generally, G is poor during early learning, and the generated examples clearly substantially differ from the training data. Therefore, D can reject these early generated examples with high confidence. In this situation, $\log(1 - D(G(z)))$ saturates. However, we can train G to maximize $\log(D(G(z)))$

rather than minimize $\log(1 - D(G(z)))$. The cost for the generator then becomes

$$J^{(G)} = E_{z\sim p_z(z)}[-\log(D(G(z)))] \\ = E_{x\sim p_g}[-\log(D(x))]. \tag{7}$$

This new objective function results in the same fixed point in the dynamics of D and G but provides much larger gradients during the early learning process. The non-saturating game is heuristic and is not motivated by theory. However, the non-saturating game has other problems, such as an unstable numerical gradient for training G. With the optimal D_G^*, we have

$$E_{x\sim p_g}\left[-\log\left(D_G^*(x)\right)\right] + E_{x\sim p_g}\left[\log\left(1 - D_G^*(x)\right)\right] \\ = E_{x\sim p_g}\left[\log\frac{\left(1 - D_G^*(x)\right)}{D_G^*(x)}\right] = E_{x\sim p_g}\left[\log\frac{p_g(x)}{p_{data}(x)}\right] \\ = KL(p_g\|p_{data}). \tag{8}$$

Therefore, $E_{x\sim p_g}\left[-\log\left(D_G^*(x)\right)\right]$ is equal to

$$E_{x\sim p_g}\left[-\log\left(D_G^*(x)\right)\right] \\ = KL(p_g\|p_{data}) - E_{x\sim p_g}\left[\log\left(1 - D_G^*(x)\right)\right]. \tag{9}$$

From (3) and (6), we have

$$E_{x\sim p_{data}}\left[\log D_G^*(x)\right] + E_{x\sim p_g}\left[\log\left(1 - D_G^*(x)\right)\right] \\ = 2JS(p_{data}\|p_g) - 2\log 2. \tag{10}$$

Therefore, $E_{x\sim p_g}\left[\log\left(1 - D_G^*(x)\right)\right]$ equals

$$E_{x\sim p_g}\left[\log\left(1 - D_G^*(x)\right)\right] \\ = 2JS(p_{data}\|p_g) - 2\log 2 - E_{x\sim p_{data}}\left[\log D_G^*(x)\right]. \tag{11}$$

By substituting (11) into (9), (9) reduces to

$$E_{x\sim p_g}\left[-\log\left(D_G^*(x)\right)\right] \\ = KL(p_g\|p_{data}) - 2JS(p_{data}\|p_g) \\ + E_{x\sim p_{data}}\left[\log D_G^*(x)\right] + 2\log 2. \tag{12}$$

From (12), we can see that optimizing the alternative G loss in the non-saturating game is contradictory because the first term aims to minimize the divergence between the generated distribution and the real distribution while the second term aims to maximize the divergence between these two distributions due to the negative sign. This results in an unstable numerical gradient when training G. Furthermore, the KL divergence is not a symmetrical quantity, as reflected by the following two examples:

- If $p_{data}(x) \to 0$ and $p_g(x) \to 1$, we have $KL(p_g\|p_{data}) \to +\infty$.
- If $p_{data}(x) \to 1$ and $p_g(x) \to 0$, we have $KL(p_g\|p_{data}) \to 0$.

The penalties for the two types of errors made by G are completely different. The first error type occurs when G produces implausible examples, which results in a large penalty. The second error type occurs when G does not produce real examples, and the penalization is quite small. The first error type involves generated examples that are inaccurate, while the second error type involves insufficiently diverse generated examples. Based on this, G will prefer to produce repetitious but safe examples rather than risk producing different but unsafe examples. This problem is termed the mode collapse problem.

3.1.1.3 Maximum likelihood game: Many methods exist to approximate (1) in GANs. Under the assumption that the discriminator is optimal, minimizing

$$J^{(G)} = E_{z\sim p_z(z)}\left[-\exp\left(\sigma^{-1}(D(G(z)))\right)\right] \\ = E_{z\sim p_z(z)}[-D(G(z))/(1 - D(G(z)))], \tag{13}$$

where σ is the logistic sigmoid function, is equal to minimizing (1) [111]. A demonstration of this equivalence can be found in Section 8.3 of [91]. Furthermore, there are other possible ways of approximating maximum likelihood within the GAN framework [20]. A comparison of the original zero-sum game, non-saturating game, and maximum likelihood game is shown in Fig. 1.

Three observations can be obtained from Fig. 1.

- First, when the example is fake (the left end of the figure), both the maximum likelihood game and the original minimax game suffer from the vanishing gradient problem. The heuristically motivated non-saturating game does not have this problem.
- Second, the maximum likelihood game also has the problem that almost all of the gradient occurs at the right end of the curve, which means that a rather small number of examples in each mini-batch dominate the gradient computation. This demonstrates that variance reduction methods based on the maximum likelihood game could be an important research direction for improving GAN performance.
- Third, the heuristically motivated non-saturating game has lower example variance, which is one possible reason why it is more successful in real applications.

GAN Lab [112] was proposed as an interactive visualization tool designed for non-experts to learn and experiment with GANs. Bau *et al.* [113] presented an analytic framework for visualizing and understanding GANs.

3.2 GAN Representative Variants

There are many papers related to GANs [114], [115], [116], [117], [118], [119], [120], [121], [122], [123], [124], [125], [126], such as least squares GAN (LSGAN) [23], cyclic-synthesized GAN (CSGAN) [127], and latent optimisation for GAN (LOGAN) [128]. In this subsection, we will introduce the representative GAN variants.

3.2.1 InfoGAN

Rather than utilizing a single unstructured noise vector z, decomposing the input noise vector into two parts was proposed for information maximizing GAN (InfoGAN) [17]: z, which is considered incompressible noise, and c, which is called the latent code and targets the significant structured semantic features of the real data distribution. InfoGAN [17] aims to solve

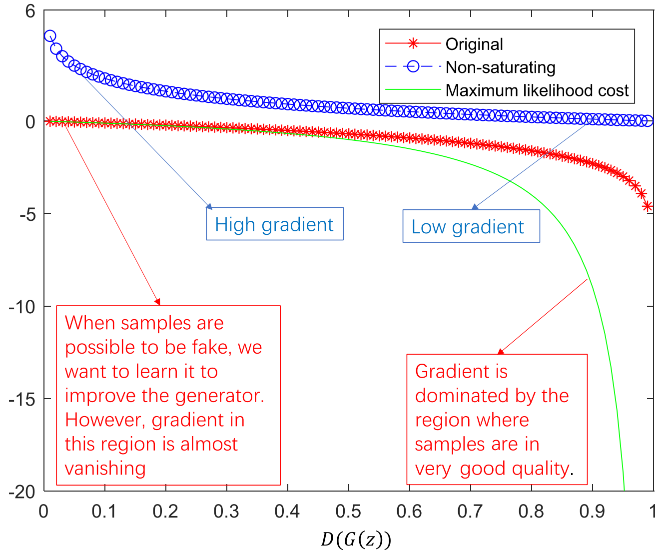

Fig. 1. The three curves for "Original", "Non-saturating", and "Maximum likelihood cost" denote $\log(1 - D(G(z)))$, $-\log(D(G(z)))$, and $-D(G(z))/(1 - D(G(z)))$ in (1), (7), and (13), respectively. The cost to the generator when generating an example $G(z)$ is determined only by the discriminator response to that generated example. The larger the probability that the discriminator gives the real label to the generated example, the smaller the cost that the generator bears. This figure is reproduced from [91], [111].

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c; G(z,c)), \tag{14}$$

where $V(D,G)$ is the objective function of the original GANs, $G(z,c)$ is the generated example, I is the mutual information, and λ is a tunable regularization parameter. Maximizing $I(c; G(z,c))$ maximizes the mutual information between c and $G(z,c)$, causing c to contain as many important and meaningful features of the real examples as possible. In practice, however, $I(c; G(z,c))$ is difficult to optimize directly since this requires access to the posterior $P(c|x)$. Fortunately, we can obtain a lower bound of $I(c; G(z,c))$ by defining an auxiliary distribution $Q(c|x)$ to approximate $P(c|x)$. The final objective function of InfoGAN [17] is

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda L_I(c; Q), \tag{15}$$

where $L_I(c; Q)$ is the lower bound of $I(c; G(z,c))$. InfoGAN has several variants, such as causal InfoGAN [129] and semi-supervised InfoGAN (ss-InfoGAN) [130].

3.2.2 Conditional GANs (cGANs)

GANs can be extended to a conditional model if both the discriminator and generator are conditioned on extra information y. The objective function of conditional GANs [18] is

$$\begin{aligned}\min_G \max_D V(D,G) = & E_{x\sim p_{data}(x)}[\log D(x|y)] \\ & + E_{z\sim p_z(z)}[\log(1 - D(G(z|y)))].\end{aligned} \tag{16}$$

By comparing (15) and (16), we can see that the InfoGAN generator is similar to that of cGANs. However, the latent code c of InfoGAN is not known; it is discovered through training. Furthermore, InfoGAN has an additional network Q to output the conditional variables $Q(c|x)$.

Based on cGANs, we can generate examples conditioned on class labels [33], [131], text [37], [132], [133], bounding boxes and keypoints [134]. In [37], [135], text to photo-realistic image synthesis was conducted with stacked generative adversarial networks (SGAN) [136]. Various cGANs have also been used for convolutional face generation [137], face aging [138], image translation [139], outdoor image synthesis with specific scenery attributes [140], natural image description [141], and 3D-aware scene manipulation [142]. Chrysos *et al.* [143] proposed robust cGANs. Thekumparampil *et al.* [144] discussed the robustness of conditional GANs to noisy labels. Conditional CycleGAN [19] used cGANs with cyclic consistency. Mode seeking GANs (MSGANs) [145] were proposed with a simple yet effective regularization term to address the mode collapse issue for cGANs.

The discriminator of original GANs [6] is trained to maximize the log-likelihood that it has assigned an example to the correct source [33]:

$$\begin{aligned}L = & E[\log P(S = real|X_{real})] \\ & + E\left[\log\left(P(S = fake|X_{fake})\right)\right],\end{aligned} \tag{17}$$

which is equivalent to (1). In contrast, the objective function of the auxiliary classifier GAN (AC-GAN) [33], [146] has two parts: the log-likelihood of the correct source, L_S, and the log-likelihood of the correct class label, L_C. Note that L_S is equivalent to L in (17). L_C is defined as following:

$$\begin{aligned}L_C = & E[\log P(C = c|X_{real})] \\ & + E\left[\log\left(P(C = c|X_{fake})\right)\right].\end{aligned} \tag{18}$$

The discriminator and generator of AC-GAN maximize $L_C + L_S$ and $L_C - L_S$, respectively. AC-GAN was the first GAN variant that was able to produce recognizable examples of all ImageNet [147] classes.

The discriminators of most cGANs-based methods [34], [44], [148], [149], [150] add conditional information y into the discriminator by simply concatenating (embedded) y with the input or with the feature vector in some middle layer. cGANs with a projection discriminator [151] adopt an inner product between the condition vector y and the feature vector.

Isola *et al.* [152] used cGANs and sparse regularization for image-to-image translation. The corresponding software is called pix2pix. In GANs, the generator learns a mapping from random noise z to $G(z)$. In contrast, no noise is input to the generator of pix2pix. One novel aspect of pix2pix is that its generator learns a mapping from an observed image y and outputs image $G(y)$, for example, from a grayscale image to a color image. In [152], the objective of cGANs is expressed as following:

$$\begin{aligned}L_{cGANs}(D,G) = & E_{x,y}[\log D(x,y)] \\ & + E_y[\log(1 - D(y, G(y)))].\end{aligned} \tag{19}$$

Furthermore, the l_1 distance is used

$$L_{l_1}(G) = E_{x,y}\left[\|x - G(y)\|_1\right]. \tag{20}$$

The final objective of [152] is

$$L_{cGANs}(D,G) + \lambda L_{l_1}(G), \tag{21}$$

where λ is the free parameter.

As a follow-up to pix2pix, pix2pixHD [153] used cGANs and feature matching loss for high-resolution image synthesis and semantic manipulation. For the discriminators, the learning problem is a multi-task learning problem:

$$\min_{G} \max_{D_1,D_2,D_3} \sum_{k=1,2,3} L_{GAN}(G, D_k). \tag{22}$$

The training set consists of pairs of corresponding images $\{(s_i, x_i)\}$, where x_i is a natural photo and s_i is a corresponding semantic label map. The ith-layer feature extractor of discriminator D_k is denoted as $D_k^{(i)}$ (from the input to the ith layer of D_k). The feature matching loss $L_{FM}(G, D_k)$ is

$$L_{FM}(G, D_k) = E_{(s,x)} \sum_{i=1}^{T} \frac{1}{N_i}\left[\left\| D_k^{(i)}(s,x) - D_k^{(i)}(s, G(s))\right\|_1\right], \tag{23}$$

where N_i is the number of elements in each layer and T denotes the total number of layers. The final objective function of [153] is

$$\min_{G} \max_{D_1,D_2,D_3} \sum_{k=1,2,3} (L_{GAN}(G, D_k) + \lambda L_{FM}(G, D_k)). \tag{24}$$

3.2.3 CycleGAN

Image-to-image translation is a class of graphics and vision problems in which the goal is to learn a mapping between an input image and an output image using a training set of aligned image pairs. When paired training data are available, reference [152] can be used for such image-to-image translation tasks. However, reference [152] cannot be used for unpaired data (when no input/output pairs are available); instead, this problem was well solved by cycle-consistent GANs (CycleGAN) [154]. CycleGAN is an important model for unpaired data. The cycle consistency was proven to be an upper bound of the conditional entropy [155]. CycleGAN is derived as a special case within the proposed variational inference (VI) framework [156], which naturally establishes its relationship with approximate Bayesian inference methods.

The basic ideas of learning to discover cross-domain relations with GANs (DiscoGAN) [157] and CycleGAN [154] are nearly the same. Both models were proposed separately at nearly the same time. The only difference between CycleGAN [154] and DualGAN [158] is that DualGAN uses the loss format advocated by the Wasserstein GAN (WGAN) rather than the sigmoid cross-entropy loss used in CycleGAN.

3.2.4 Summary

The website "The GAN Zoo" (https://github.com/hindupuravinash/the-gan-zoo) lists many GAN variants. Please refer to this website for more details.

3.3 GAN Training

Despite the theoretical existence of unique solutions, GAN training is difficult and often unstable for several reasons [32], [35], [159]. One difficulty stems from the fact that the optimal weights for GANs correspond to saddle points rather than minima of the loss function.

Many papers exist that focus on GAN training. Yadav *et al.* [160] stabilized GAN training using prediction methods. By using independent learning rates, [161] proposed a two timescale update rule (TTUR) for both the discriminator and generator to ensure that the model would converge to a stable local Nash equilibrium. Arjovsky [159] took theoretical steps toward fully understanding the training dynamics of GANs; analyzed why GANs are difficult to train; studied and proved several problems including saturation and instability, that can occur when training GANs; examined a practical and theoretically grounded direction to mitigate these problems; and introduced new tools to study them.

One approach to improving GAN training is to assess the empirical "symptoms" that might occur in training. These symptoms include mode collapse (*cf.* Subsection 4.2); the discriminator loss converging quickly to zero [159] and providing no gradient updates to the generator; and difficulties in making both the generator and discriminator converge [35].

Here, we introduce GAN training from three perspectives: objective function, skills, and structure.

3.3.1 Objective Function

As discussed in Subsection 3.1, utilizing the original objective function in Eq. (1) can cause the vanishing gradient problem when training G, while utilizing the alternative G loss (12) in the non-saturating game can result in the mode collapse problem. These problems are directly caused by the objective function and cannot be solved by changing the GAN structure. Re-designing the objective function is a natural solution to mitigate these problems. Based on the theoretical flaws of GANs, many objective function based variants have been proposed that change the objective function of GANs based on theoretical analyses, such as least squares generative adversarial networks [24], [25]. Lucic *et al.* [162] conducted a large-scale experimental study and found that no GAN variant consistently outperformed the original GANs. Next, we introduce a series of objective function based variants.

3.3.1.1 Least squares generative adversarial networks (LSGANs): LSGANs [24], [25] were proposed to overcome the vanishing gradient problem in the original GANs. The decision boundary for D of original GANs was shown to provide only very small penalties to update G when generated examples are far from the decision boundary. Thus, LSGANs adopt least squares loss rather than the cross-entropy loss used in the original GANs. Suppose that a-b coding is used for the LSGANs discriminator [24], where a and b are the labels for the generated and real examples, respectively. The LSGANs discriminator loss $V_{LSGAN}(D)$ and generator loss $V_{LSGAN}(G)$ are defined as follows:

$$\begin{aligned}\min_{D} V_{LSGAN}(D) = {} & E_{x\sim p_{data}(x)}\left[(D(x)-b)^2\right] \\ & + E_{z\sim p_z(z)}\left[(D(G(z))-a)^2\right],\end{aligned} \tag{25}$$

$$\min_G V_{LSGAN}(G) = E_{z\sim p_z(z)}\Big[(D(G(z)) - c)^2\Big], \tag{26}$$

where c is the value that G hopes for D to believe for generated examples. The authors of [24] showed that LSGANs have two advantages over the original GANs.

- The new decision boundary produced by D imposes a large penalty for generated examples that are far from the decision boundary, which forces the "low quality" generated examples to move toward the decision boundary. This approach is effective at generating higher quality examples.
- Penalizing generated examples far from the decision boundary results in larger gradients when updating G, which overcomes the vanishing gradient problems in the original GANs.

3.3.1.2 f-GAN: The KL divergence measures the difference between two probability distributions. A large class of assorted divergences are the so-called Ali-Silvey distances, also known as the f-divergences [163]. Given two probability distributions P and Q that have absolutely continuous density functions p and q, respectively, with regard to a base measure dx defined on domain X, the f-divergence is defined as following:

$$D_f(P\|Q) = \int_X q(x) f\left(\frac{p(x)}{q(x)}\right) dx. \tag{27}$$

Different choices of f recover popular divergences as special cases of the f-divergence. For example, if $f(a) = a \log a$, the f-divergence becomes the KL divergence. The original GANs [6] are a special case of f-GAN [20], which is based on the f-divergence. The authors of [20] showed that any f-divergence can be used for training GANs. Furthermore, [20] discussed the advantages of different choices of divergence functions on both the quality of the produced generative models and the training complexity. Im *et al.* [164] quantitatively evaluated GANs with divergences proposed for training. Uehara *et al.* [165] further extended f-GAN by directly minimizing the f-divergence in the generator step; then, the ratio of the real and generated data distributions are predicted in the discriminator step.

3.3.1.3 Integral probability metrics (IPMs): $\mathcal{P}$ denotes the set of all Borel probability measures on a topological space $(M, \mathcal{A})$. The integral probability metric (IPM) [166], [167] between two probability distributions $P \in \mathcal{P}$ and $Q \in \mathcal{P}$ is defined as

$$\gamma_{\mathcal{F}}(P,Q) = \sup_{f\in\mathcal{F}}\left|\int_M f dP - \int_M f dQ\right|, \tag{28}$$

where $\mathcal{F}$ is a class of real-valued bounded measurable functions on M. IPMs include the reproducing kernel Hilbert space (RKHS)-induced maximum mean discrepancy (MMD) [168] and the Wasserstein distance used in WGAN.

MMD: The following definition of the MMD can be found in [169]. Here, $\mathcal{X}$ represents the input domain, which is assumed to be a nonempty compact set.

Definition 1. *Let $\mathcal{E}$ be a class of functions $f : \mathcal{E} \to R$. Let P and Q be Borel probability distributions, and let $X = (x_1, \cdots, x_m)$ and $Y = (y_1, \cdots, y_n)$ be examples consisting of independent and identically distributed observations drawn from P and Q, respectively. Then, the MMD and its empirical estimate are defined as follows:*

$$\begin{aligned} MMD(\mathcal{E},P,Q) &= \sup_{f\in\mathcal{E}}\big(E_{x\sim P}[f(x)] - E_{y\sim Q}[f(y)]\big) MMD(\mathcal{E},X,Y) \\ &= \sup_{f\in\mathcal{E}}\left(\frac{1}{m}\sum_{i=1}^{m} f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(y_i)\right). \end{aligned} \tag{29}$$

When $\mathcal{E}$ is the unit ball in a universal RKHS, Theorem 2.2 in [169] guarantees that $MMD(\mathcal{E},P,Q)$ will detect any discrepancy between P and Q. The MMD has been widely used for GANs [170], [171], [172], [173], [174], [175], [176], [177].

WGAN: The authors of [21] conducted a comprehensive theoretical analysis of how the Wasserstein-1 distance behaves in comparison with popular probability distances and divergences such as the total variation (TV) distance, the KL divergence, and the JS divergence utilized in the context of learning distributions. The definition of the Wasserstein-1 distance is

$$W(p_{data}, p_g) = \inf_{\gamma\in\Pi(p_{data},p_g)} E_{(x,y)\in\gamma}[\|x-y\|], \tag{30}$$

where $\Pi(p_{data}, p_g)$ denotes the set of all joint distributions $\gamma(x,y)$ whose marginals are p_{data} and p_g. However, the infimum in (30) is highly intractable. According to the Kantorovich-Rubinstein duality [178], we know that

$$W(p_{data}, p_g) = \sup_{\|f\|_L\le 1} E_{x\in p_{data}}[f(x)] - E_{x\in p_g}[f(x)] \tag{31}$$

where the supremum is taken over all the 1-Lipschitz functions f. In [21], $\|f\|_L \le 1$ was replaced with $\|f\|_L \le K$ (considering K-Lipschitz for some constant K), and $K \cdot W(p_{data}, p_g)$ was obtained. The authors of [21] used the following equation to approximate the Wasserstein-1 distance:

$$\max_{w\in\mathcal{W}} E_{x\sim p_{data}(x)}[f_w(x)] - E_{z\sim p_z(z)}[f_w(G(z))], \tag{32}$$

where a parameterized family of functions $\{f_w\}_{w\in\mathcal{W}}$ exists that are all K-Lipschitz for some K, and f_w can be realized by the discriminator D. When D is optimized, (32) denotes the approximated Wasserstein-1 distance. Then, the aim of G is to minimize (32) to make the generated distribution as close to the real distribution as possible. Therefore, the overall objective function of WGAN is

$$\begin{aligned} &\min_G \max\nolimits_{w\in\mathcal{W}} E_{x\sim p_{data}(x)}[f_w(x)] - E_{z\sim p_z(z)}[f_w(G(z))] \\ &\quad = \min_G \max_D E_{x\sim p_{data}(x)}[D(x)] - E_{z\sim p_z(z)}[D(G(z))]. \end{aligned} \tag{33}$$

By comparing (1) and (33), we can see three differences between the objective function of the original GANs and that of the WGAN:

- First, there is no *log* in the objective function of WGAN.

- Second, the D in the original GANs is utilized as a binary classifier, while the D in WGAN is utilized to approximate the Wasserstein distance, which is a regression task. Therefore, the sigmoid function that appears in the last layer of D is not used in WGAN; the discriminator of the original GANs outputs a value between zero and one, while no such constraint exists for WGAN.
- Third, the D in WGAN is required to be K-Lipschitz for some K; therefore, WGAN uses weight clipping.

Compared with traditional GAN training, WGAN improves the learning stability and provides meaningful learning curves that are useful for hyperparameter searches and debugging. However, approximating the K-Lipschitz constraint is challenging, which is required by the Wasserstein-1 metric. WGAN-GP, proposed in [22], uses a gradient penalty to restrict the K-Lipschitz constraint, and the WGAN-GP objective function is

$$\begin{aligned} L = & -E_{x\sim p_{data}}[D(x)] + E_{\tilde{x}\sim p_g}[D(\tilde{x})] \\ & +\lambda E_{\hat{x}\sim p_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1\right)^2\right] \end{aligned} \quad (34)$$

where the first two terms are the WGAN objective function and $\hat{x}$ is sampled from the distribution $p_{\hat{x}}$; these are uniform examples along straight lines between pairs of points sampled from the real data distribution p_{data} and the generated distribution p_g. Gradient penalties are now a commonly used approach in GANs, following [179], [180], [181]. Some other methods are closely related to WGAN-GP, such as deep regret analytic GAN (DRAGAN) [182]. Wu *et al.* [183] proposed a novel and relaxed version of the Wasserstein-1 metric called the Wasserstein divergence (W-div), which does not require the K-Lipschitz constraint. Based on W-div, Wu *et al.* [183] introduced a Wasserstein divergence objective for GANs (WGAN-div) that faithfully approximates W-div through optimization. The Wasserstein distance was argued to lead to biased gradients, and the use of the Cramér distance between two distributions was suggested and implemented in CramerGAN [184]. Other papers related to WGAN can be found in [185], [186], [187], [188], [189].

3.3.1.4: Spectrally normalized GANs (SN-GANs): A novel weight normalization method named spectral normalization to stabilize the discriminator training was proposed in SN-GANs [26]. This new normalization technique is both computationally efficient and easy to integrate into existing methods. Spectral normalization [26] uses a simple method to make the weight matrix W satisfy the Lipschitz constraint $\sigma(W) = 1$:

$$\bar{W}_{SN}(W) := W/\sigma(W), \quad (35)$$

where W is the weight matrix of each layer in D, and $\sigma(W)$ is the spectral norm of W. As shown in [26], SN-GANs can generate images of equal or better quality than the previous training stabilization methods. In theory, spectral normalization can be applied to all GAN variants. Both BigGANs [39] and self-attention GAN (SAGAN) [38] use spectral normalization and have achieved good performances on ImageNet.

3.3.1.5 Relativistic GANs (RGANs): In the original GANs, the discriminator can be defined according to the non-transformed layer $C(x)$ as $D(x) = \sigma(C(x))$. A simple way to make the discriminator relativistic (i.e., to make the output of D depend on both real and generated examples) [31] is to sample from real and generated data pairs $\tilde{x} = (x_r, x_g)$, which is defined as

$$D(\tilde{x}) = \sigma\big(C(x_r) - C(x_g)\big). \quad (36)$$

This modification can be interpreted in the following way [31]: D estimates the probability that the given real example is more realistic than a randomly sampled generated example. Similarly, $D_{rev}(\tilde{x}) = \sigma\big(C(x_g) - C(x_r)\big)$ can be interpreted as the probability that the given generated example is more realistic than a randomly sampled real example. The discriminator and generator loss functions of the relativistic standard GAN (RSGAN) are

$$L_D^{RSGAN} = -E_{(x_r,x_g)}\left[\log\left(\sigma\big(C(x_r) - C(x_g)\big)\right)\right], \quad (37)$$

$$L_G^{RSGAN} = -E_{(x_r,x_g)}\left[\log\left(\sigma\big(C(x_g) - C(x_r)\big)\right)\right]. \quad (38)$$

Most GANs can be parameterized

$$L_D^{GAN} = E_{x_r}[f_1(C(x_r))] + E_{x_g}\big[f_2\big(C(x_g)\big)\big], \quad (39)$$

$$L_G^{GAN} = E_{x_r}[g_1(C(x_r))] + E_{x_g}\big[g_2\big(C(x_g)\big)\big], \quad (40)$$

where f_1, f_2, g_1, and g_2 are scalar-to-scalar functions. If we adopt a relativistic discriminator, the loss functions of these GANs become

$$\begin{aligned} L_D^{RGAN} = & E_{(x_r,x_g)}\big[f_1\big(C(x_r) - C(x_g)\big)\big] \\ & + E_{(x_r,x_g)}\big[f_2\big(C(x_g) - C(x_r)\big)\big] \end{aligned}, \quad (41)$$

$$\begin{aligned} L_G^{RGAN} = & E_{(x_r,x_g)}\big[g_1\big(C(x_r) - C(x_g)\big)\big] \\ & + E_{(x_r,x_g)}\big[g_2\big(C(x_g) - C(x_r)\big)\big]. \end{aligned} \quad (42)$$

3.3.2 Skills

NIPS 2016 held a workshop on adversarial training and invited Soumith Chintala to give a talk called "How to train a GAN". This talk included assorted tips and tricks, such as suggesting that when labels are available, also training the discriminator to classify the examples is useful, as in AC-GAN [33]. Readers can refer to the GitHub repository associated with Soumith's talk, https://github.com/soumith/ganhacks, for more advice.

Salimans *et al.* [32] proposed useful and improved techniques for training GANs (ImprovedGANs), such as feature matching, mini-batch discrimination, historical averaging, one-sided label smoothing, and virtual batch normalization.

3.3.3 Structure

The original GANs utilized multi-layer perceptrons (MLPs). Specific types of structures may be better for specific applications, e.g., recurrent neural networks (RNNs) for time series data and convolutional neural networks (CNNs) for images.

3.3.3.1 The original GANs: The original GANs used MLPs for both the generator G and discriminator D.

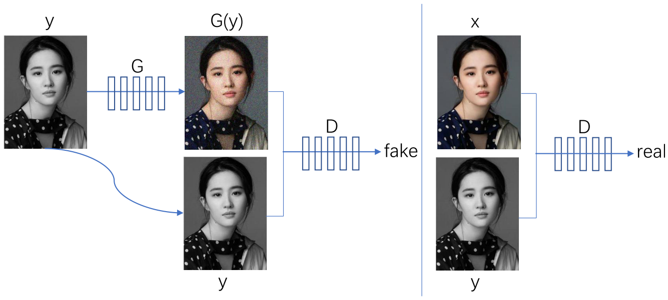

Fig. 2. pix2pix: training conditional GANs to map grayscale images → color images. The discriminator D learns to classify between real {grayscale, color} and fake (synthesized by the generator) tuples. The generator G learns to fool the discriminator. Different from the original GANs, both the generator and discriminator receive the grayscale image as input, and the pix2pix generator receives no noise input.

However, an MLP can be used only for small datasets such as CIFAR-10 [190], MNIST [191], and the Toronto Face Database (TFD) [192]; it does not generalize well to more complex images [14].

3.3.3.2 Deep convolutional generative adversarial networks (DCGANs): In the original GANs, G and D are defined using an MLP. Because CNNs are better at images than MLPs, G and D are defined by deep convolutional neural networks (DCNNs) in DCGANs [35], which have better performance. Most current GANs are at least loosely based on the DCGAN architecture [35]. The three key features of the DCGAN architecture are listed as follows.

- First, the overall architecture is based largely on an all-convolutional net [193], which includes neither pooling nor "unpooling" layers. When G needs to increase the spatial dimensionality of the representation, it uses transposed convolution (deconvolution) with a stride greater than 1.
- Second, DCGANs use batch normalization for most of the layers of both G and D. The last layer of G and first layer of D are not batch normalized, which allows the neural network to learn the correct mean and scale of the data distribution.
- Third, DCGANs use the adaptive moment estimation (Adam) optimizer instead of stochastic gradient descent (SGD) with momentum.

3.3.3.3 Progressive GAN: A new training methodology for GANs was proposed and implemented in progressive GAN (PGGAN) [36]. The structure of PGAN is based on progressive neural networks, which were first proposed in [194]. The key idea of PGAN is to train both the generator and discriminator progressively: starting at a low resolution and adding new layers that model increasingly fine details as training progresses.

3.3.3.4 Self-Attention Generative Adversarial Network (SAGAN): SAGAN [38] was proposed to allow attention-driven, long-range dependency modeling for image generation tasks. The spectral normalization technique had previously been applied only to the discriminator [26], but SAGAN uses spectral normalization for both the generator and discriminator, which improves the training dynamics. Furthermore, SAGAN confirmed that TTUR [161] was effective.

Note that AttnGAN [195] utilizes attention over word embeddings within an input sequence rather than self-attention over internal model states.

3.4 Summary

As discussed, many GAN variants have been constructed; some of the milestone variants are shown in Fig. 3. Note that due to space limitations, only a limited number of variants are shown.

The objective function based variants of GANs can be generalized to structure variants. Compared with other objective function based variants, both SN-GANs and RGANs show stronger generalization ability. These two objective function based variants can be generalized to the other objective function based variants. Spectral normalization can be generalized to any type of GAN variant, while RGANs can be generalized to any IPM-based GANs.

Objective function based variants such as energy-based generative adversarial network (EBGAN) [41] and boundary equilibrium generative adversarial networks (BEGAN) [42], structure variants such as Laplacian generative adversarial networks (LAPGAN) [34] and SinGAN [196], evaluation metrics for GANs, and task-driven GANs are discussed in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2021.3130191.

4 Theory

In this section, we first introduce maximum likelihood estimation. Then, we introduce mode collapse. Finally, we discuss other theoretical issues, such as memorization.

4.1 Maximum Likelihood Estimation (MLE)

Not all generative models use MLE. Some generative models do not utilize MLE but can be made to do so (GANs belong to this category). Minimizing the KL divergence between $p_{data}(x)$ and $p_g(x)$ can be simply proven to be

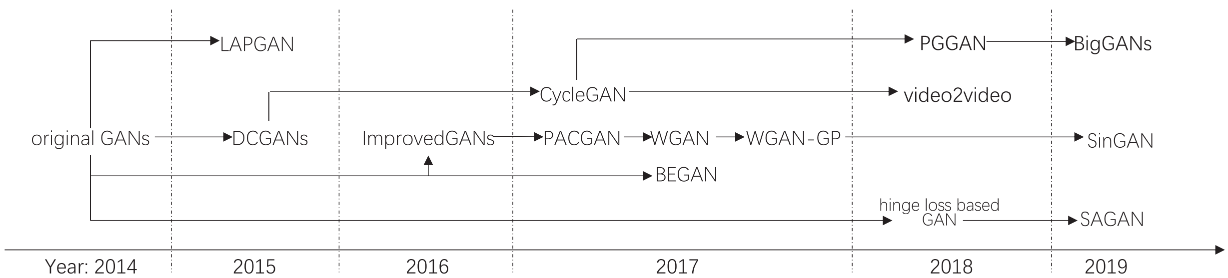

Fig. 3. Road map of GANs showing milestone variants.

equivalent to maximizing the log-likelihood as the number of examples m increases:

$$\begin{aligned}\theta^* &= \arg\min_\theta \ KLD(p_{data}\|p_g)\\ &= \arg\min_\theta \ -\int p_{data}(x)\log\frac{p_g(x)}{p_{data}(x)}dx\\ &= \arg\min_\theta \int p_{data}(x)\log p_{data}(x)dx\\ &\quad -\int p_{data}(x)\log p_g(x)dx\\ &= \arg\max_\theta \int p_{data}(x)\log p_g(x)dx\\ &= \arg\max_\theta \ \lim_{m\to\infty}\frac{1}{m}\sum\nolimits_{i=1}^{m}\log p_g(x_i).\end{aligned} \tag{43}$$

The model probability distribution $p_\theta(x)$ is replaced with $p_g(x)$ for notation consistency. Refer to Chapter 5 of [197] for more information on MLE and other statistical estimators.

4.2 Mode Collapse

GANs are notoriously difficult to train, and they have been observed [29], [32] to suffer from mode collapse [198], [199], in which the generator learns to generate examples from only a few modes of the data distribution and misses many other modes, even if examples of the missing modes exist throughout the training data. In the worst case, the generator simply produces a single example (complete collapse) [159], [200]. In this subsection, we first introduce two viewpoints regarding GAN mode collapse. Then, we introduce methods that propose new objective functions or new structures to solve the mode collapse problem.

4.2.1 Two Viewpoints: Divergence and Algorithmic

We can resolve and understand GAN mode collapse and instability from both divergence and algorithmic viewpoints.

Divergence Viewpoint. Roth *et al.* [179] stabilized the training of GANs and their variants, such as f-divergence based GANs (f-GAN), through regularization.

Algorithmic Viewpoint. The numerics of common algorithms for training GANs were analyzed and a new algorithm with better convergence was proposed in [201]. Mescheder *et al.* [180] showed which training methods for GANs actually converge.

4.2.2 Methods to Overcome Mode Collapse

Objective Function Based Methods. Mode collapse was suggested to occur because of a fake local Nash equilibrium in the non-convex problem [182]. DRAGAN solved this problem by constraining the gradients of the discriminator around the real data manifold. It added a gradient penalizing term that biases the discriminator to have a gradient norm of 1 around the real data manifold. Other methods, such as LSGANs and SN-GANs (detailed in Section 3.3), also belong to this category.

Structure-Based Methods. The representative methods in this category include DCGANs and SAGAN (detailed in Section 3.3).

Other methods also exist that can reduce mode collapse in GANs. For example, PACGAN [202] alleviated mode collapse by changing the input to the discriminator.

4.3 Other Theoretical Issues

4.3.1 Do GANs Actually Learn the Distribution?

Perhaps the most crucial aspect of GAN theory is whether the distributions are modeled. Both true data distributions and GAN generator distributions have their own densities. As [159] noted, neither distribution typically has a density in GANs. Furthermore, [159] studied and proved the problems involved in training GANs, such as saturation and instability, investigated directions to mitigate these problems and introduced new tools to study them.

Several studies [44], [200], [203] both empirically and theoretically shed light on the fact that distributions learned by GANs suffer from mode collapse. In contrast, Bai *et al.* [204] showed that GANs can in principle learn distributions using the Wasserstein distance (or KL divergence in many situations) with polynomial sample complexity if the discriminator class has strong discriminating power against the particular generator class (instead of against all possible generators). Liang *et al.* [205] studied how well GANs learn densities, including nonparametric and parametric target distributions. Singh *et al.* [206] further studied nonparametric density estimation with adversarial losses.

4.3.2 Divergence/Distance

Arora *et al.* [200] showed that GAN training may not result in good generalization properties (e.g., training may look successful, but the generated distribution may be far from the real data distribution using standard metrics). Popular distances such as Wasserstein and JS may not generalize well. However, generalization can still occur by introducing a novel notion of distance between distributions—the neural net distance—which raises the issue of whether other useful divergences exist.

4.3.3 Mathematical Perspectives Such as Optimization

Mohamed *et al.* [207] used their understanding of GANs to build connections to the diverse set of statistical thinking regarding GANs. Gidel *et al.* [208] examined optimization approaches designed for GANs and cast GAN optimization problems into the general variational inequality framework. The convergence and robustness of training GANs with regularized optimal transport are discussed in [209].

4.3.4 Memorization

Regarding "memorization of GANs", Nagarajan *et al.* [210] argued that making the generator "learn to memorize" the training data is more difficult than making it "learn to output realistic but unseen data".

5 APPLICATIONS

As discussed earlier, GANs are powerful generative models that can generate realistic-looking examples with a random vector z. GANs neither need to know an explicit true data distribution nor require any prior mathematical assumptions. These advantages allow GANs to be widely applied

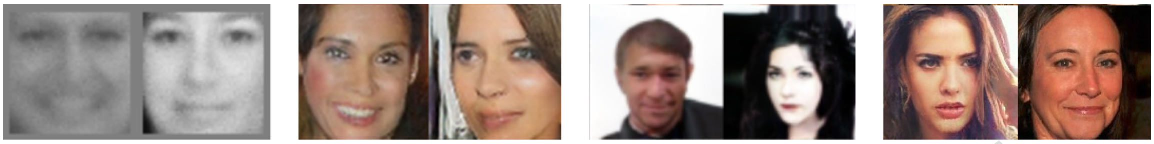

Fig. 4. Face image synthesis.

to many fields, such as image processing, computer vision, and sequential data.

5.1 Image Processing and Computer Vision

The most successful applications of GANs are in image processing and computer vision, such as image super-resolution, image synthesis and manipulation, and video processing.

5.1.1 Super-Resolution (SR)

SRGAN [51], which is a GAN model for performing SR, was the first framework able to infer photo-realistic natural images for 4× upscaling factors. To further improve the visual quality of SRGAN, Wang *et al.* [52] thoroughly studied three of its key components and improved each one to derive an enhanced SRGAN (ESRGAN). For example, ESRGAN uses the idea from relativistic GANs [31] of having the discriminator predict relative realness rather than absolute value. Benefiting from these improvements, ESRGAN won first place in the PIRM2018-SR Challenge (region 3) [211] and obtained the best perceptual index. Based on CycleGAN [154], cycle-in-cycle GANs [53] were proposed for unsupervised image SR. SRDGAN [54] were proposed to learn the noise prior for SR with DualGAN [158]. Deep tensor generative adversarial nets (TGAN) [55] were proposed to generate large high-quality images by exploring tensor structures. Specific methods have been designed for face SR [212], [213], [214]. Other related methods can be found in [215], [216], [217], [218].

5.1.2 Image Synthesis and Manipulation

5.1.2.1 Faces: *Pose related:* Disentangled representation learning GAN (DR-GAN) [219] was proposed for pose-invariant face recognition. Huang *et al.* [57] proposed a two-pathway GAN (TP-GAN) for photorealistic frontal view synthesis by simultaneously perceiving both local details and global structures. Ma *et al.* [58] proposed the novel pose guided person generation network (PG^2) that synthesizes person images in arbitrary poses based on a novel pose and an image of that person. Cao *et al.* [220] proposed a high-fidelity pose-invariant model for high-resolution face frontalization based on GANs. Siarohin *et al.* [221] proposed deformable GANs for pose-based human image generation. Pose-robust spatial-aware GAN (PSGAN) was proposed for customizable makeup transfer in [59].

Portrait Related: APDrawingGAN [60] was proposed to generate artistic portrait drawings from face photos with hierarchical GANs. APDrawingGAN has software based on WeChat, and the results are shown in Fig. 5. GANs have also been used in other face-related applications, such as facial attribute changes [222] and portrait editing [223], [224], [225], [226].

Face Generation: The quality of faces generated by GANs has steadily improved year over year; examples can be found in Sebastian Nowozin's GAN lecture materials[1]. As shown in Fig. 4, faces generated based on the original GANs [6] have poor visual qualities and serve only as a proof of concept. Radford *et al.* [35] used better neural network architectures—deep convolutional neural networks—for generating faces. Roth *et al.* [179] addressed GAN training instability problems, which allowed larger architectures such as ResNet to be utilized. Karras *et al.* [36] utilized multi-scale training to enable megapixel face image generation with high fidelity.

Face generation [19], [227], [228], [229], [230], [231], [232], [233] is relatively easy because the problem includes only one class of objects. Every object is a face, and most face datasets tend to be composed of people looking straight into the camera. Most people are registered by putting nose, eyes, and other landmarks in consistent locations.

5.1.2.2 General objects: Having GANs work on assorted data sets, such as ImageNet [147], which has a thousand different object classes, is slightly more difficult. However, progress on this task has been rapid in recent years, and the quality of such generated images has steadily improved [180].

Most studies use GANs to synthesize 2D images [234], [235]; however, Wu *et al.* [236] synthesized three-dimensional (3D) novel objects such as cars, chairs, sofas, and tables using GANs and volumetric convolutions. Im *et al.* [237] generated images with recurrent adversarial networks. Yang *et al.* [238] proposed layered recursive GANs (LR-GAN) for image generation.

5.1.2.3 Interaction between a human being and an image generation process: Many applications involve interactions

(a) photo (b) portrait drawings

Fig. 5. Given a photo such as the image in (a), APDrawingGAN can produce corresponding artistic portrait drawings such as the image in (b).

1. https://github.com/nowozin/mlss2018-madrid-gan

between a human being and an image generation process; however, realistic image manipulation in such situations is difficult because it requires allowing the user to control image modifications while still making them appear realistic. When the user does not have sufficient artistic skill, the image easily deviates from the manifold of natural images while editing. Interactive GAN (IGAN) [61] defines a class of image editing operations and constrains their output to lie on a learned manifold at all times. Introspective adversarial networks [62] also offer the ability to perform interactive photo editing; their results have been demonstrated mostly for face editing. GauGAN [63] can turn doodles into stunning, photorealistic landscapes.

5.1.3 Texture Synthesis

Texture synthesis is a classical problem in the image field. Markovian GANs (MGAN) [64] is a texture synthesis method based on GANs. By capturing the texture data of Markovian patches, MGAN can generate stylized videos and images very quickly and realize real-time texture synthesis. Spatial GAN (SGAN) [65] was the first to apply GANs with fully unsupervised learning to texture synthesis. Periodic spatial GAN (PSGAN) [66] is an SGAN variant that can learn periodic textures from either a single image or a large complex dataset.

5.1.4 Object Detection

How can we learn an object detector that is invariant to deformations and occlusions? One way is to use a data-driven strategy—collecting large-scale datasets that have numerous object examples that appear in different conditions. Using this strategy, we can simply hope that the final classifier can use these numerous instances to learn invariances. However, can all the possible deformations and occlusions be included in a dataset? Some deformations and occlusions are so rare that they almost never occur in real-world conditions; however, we want our method to be invariant to such situations. To address this problem, Wang *et al.* [239] used GANs to generate instances with deformations and occlusions. The goal of the generator is to generate instances that are difficult for the object detector to classify. By using a segmentation model and GANs, Segan [67] detected objects occluded by other objects in an image. To address the small object detection problem, Li *et al.* [68] proposed perceptual GANs, and Bai *et al.* [69] proposed an end-to-end multi-task GAN (MTGAN).

5.1.5 Video Applications

The first study to use GANs for video generation was [70]. Villegas *et al.* [240] proposed a deep neural network to predict future frames in natural video sequences using GANs. Denton and Birodkar [71] proposed a new model named disentangled representation net (DRNET) that learns disentangled image representations from a video based on GANs. A novel video-to-video synthesis approach (video2video) under a generative adversarial learning framework was proposed in [73]. MoCoGan [74] was proposed to decompose motion and content to generate videos [241], [242].

GANs have also been used in other video applications, such as video prediction [72], [243], [244] and video retargeting [245].

5.1.6 Other Image and Vision Applications

GANs have also been utilized in other image processing and computer vision tasks [246], [247], [248], such as object transfiguration [249], [250], semantic segmentation [251], visual saliency prediction [252], object tracking [253], [254], image dehazing [255], [256], [257], natural image matting [258], image inpainting [259], [260], image fusion [261], image completion [262], [263], and image classification [264].

Creswell *et al.* [265] showed that the representations learned by GANs can also be used for retrieval. GANs have also been used to anticipate where people will look next [266], [267].

5.2 Sequential Data

Additionally, GANs have made achievements in sequential data tasks, such as those involving natural language, music, speech, voice [268], [269], and time series data [270], [271], [272], [273].

Natural Language Processing (NLP). IRGAN [76], [77] was proposed for information retrieval (IR). Li *et al.* [274] used adversarial learning to generate neural dialogue. GANs have also been used for text generation [75], [275], [276], [277] and speech language processing [81]. KBGAN [278] was proposed to generate high-quality negative examples, and it was used in knowledge graph embeddings. Adversarial REward Learning (AREL) [279] was proposed for visual storytelling. DSGAN [280] was proposed for distant supervision relation extraction. ScratchGAN [281] was proposed to train a language GAN from scratch—without maximum likelihood pre-training.

Qiao *et al.* [78] learned text-to-image generation by redescription, and a text conditioned auxiliary classifier GAN (TAC-GAN) [282] was also proposed for text-to-image tasks. GANs have also been widely used for image-to-text tasks (image captioning) [283], [284].

Furthermore, GANs have been widely utilized in other NLP applications, such as question-answer selection [285], [286], poetry generation [287], talent-job fit [288], and review detection and generation [289], [290].

Music: GANs have also been used to generate music, including continuous RNN-GAN (C-RNN-GAN) [79], object-reinforced GAN (ORGAN) [80], and sequence GANs (SeqGAN) [81].

Speech and Audio. GANs have been used for speech and audio analysis, such as synthesis [291], [292], [293], enhancement [294], and recognition [295].

5.3 Other Applications

Medical Field. GANs have been widely utilized in the medical fields such as for generating and designing DNA [296], [297], drug discovery [298], generating multi-label discrete patient records [299], medical image processing [300], [301], [302], [303], [304], [305], [306], [307], and doctor recommendation [308].

Data Science. GANs have been used to generate data [309], [310], [311], [312], [313], [314], [315], [316], to generate neural networks [317], to augment data [318], [319], to learn spatial representations [320], and in network embedding [321], heterogeneous information networks [322], and mobile user profiling [323].

Finally, GANs have been widely applied to many other areas, such as malware detection [324], steganography [325], [326], [327], [328], privacy preserving [329], [330], [331], social robots [332], and network pruning [333], [334].

6 Conclusion

This paper provides a comprehensive review of various aspects of GANs by elaborating on several perspectives, i.e., algorithms, theory, and applications. We believe that this survey will help readers gain a thorough understanding of the existing research on GANs. To conclude, we would like to note that, in order to maintain an appropriate size of the article, we had to limit the number of referenced studies. We therefore apologize to the authors of papers that were not cited.

Acknowledgments

The authors would like to thank the NetEase course taught by Shuang Yang, Ian Goodfellow's invited talk at AAAI 19, the CVPR 2018 tutorial on GANs, and Sebastian Nowozin's MLSS 2018 GAN lecture materials. The authors would also like to thank Shuang Yang, Weinan Zhang, and the members of the Umich Yelab and Foreseer research groups for helpful discussions.

References

[1] L. J. Ratliff, S. A. Burden, and S. S. Sastry, "Characterization and computation of local Nash equilibria in continuous games," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, 2013, pp. 917–924.

[2] J. Schmidhuber, "Making the world differentiable: On using fully recurrent self-supervised neural networks for dynamic reinforcement learning and planning in non-stationary environments," *Inst. Comput. Sci., Tech. Univ. Munich, Germany, FKI-126, Tech. Rep.*, 1990.

[3] J. Schmidhuber, "A possibility for implementing curiosity and boredom in model-building neural controllers," in *Proc. Int. Conf. Simul. Adaptive Behav., From Animals Animats*, 1991, pp. 222–227.

[4] J. Schmidhuber, "Art & science as by-products of the search for novel patterns, or data compressible in unknown yet learnable ways," *M. Botta ed., et al. Edizioni*, pp. 98–112, 2009. [Online]. Available: https://people.idsia.ch/~juergen/onlinepub.html

[5] J. Schmidhuber, "Learning factorial codes by predictability minimization," *Neural Comput.*, vol. 4, no. 6, pp. 863–879, 1992.

[6] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Neural Inf. Process. Syst.*, pp. 2672–2680, 2014.

[7] J. Schmidhuber, "Unsupervised minimax: Adversarial curiosity, generative adversarial networks, and predictability minimization," 2020, *arXiv:1906.04493*.

[8] X. Wu, K. Xu, and P. Hall, "A survey of image synthesis and editing with generative adversarial networks," *Tsinghua Sci. Technol.*, vol. 22, no. 6, pp. 660–674, 2017.

[9] R. Zhou, C. Jiang, and Q. Xu, "A survey on generative adversarial network-based text-to-image synthesis," *Neurocomputing*, vol. 451, pp. 316–336, 2021.

[10] N. Torres-Reyes and S. Latifi, "Audio enhancement and synthesis using generative adversarial networks: A survey," *Int. J. Comput. Appl.*, vol. 182, no. 35, pp. 27–31, 2019.

[11] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.

[12] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial networks and their variants work: An overview," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–43, 2019.

[13] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[14] Z. Wang, Q. She, and T. E. Ward, "Generative adversarial networks in computer vision: A survey and taxonomy," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021.

[15] M. Zamorski, A. Zdobylak, M. Zieba, and J. Swiatek, "Generative adversarial networks: recent developments," in *Proc. Int. Conf. Artif. Intell. Soft Comput.*, 2019, pp. 248–258.

[16] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (GANs): A survey," *IEEE Access*, vol. 7, pp. 36322–36333, 2019.

[17] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[18] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[19] Y. Lu, Y.-W. Tai, and C.-K. Tang, "Attribute-guided face generation using conditional cyclegan," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 282–297.

[20] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 271–279.

[21] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[22] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[23] G.-J. Qi, "Loss-sensitive generative adversarial networks on lipschitz densities," in *Proc. Int. J. Comput. Vis.*, 2019, pp. 1–23.

[24] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2794–2802.

[25] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "On the effectiveness of least squares generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 2947–2960, Dec. 2019.

[26] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–26.

[27] J. H. Lim and J. C. Ye, "Geometric GAN," 2017, *arXiv:1705.02894*.

[28] D. Tran, R. Ranganath, and D. M. Blei, "Hierarchical implicit models and likelihood-free variational inference," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 2794–2802.

[29] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–13.

[30] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–25.

[31] A. Jolicoeur-Martineau, "The relativistic discriminator: A key element missing from standard GAN," in *Proc. Int. Conf. Learn. Representation*, 2019, pp. 1–13.

[32] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[33] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[34] E. L. Denton, S. Chintala, A. D. Szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 1486–1494.

[35] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–16.

[36] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–26.

[37] H. Zhang *et al.*, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 5907–5915.
[38] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.
[39] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–29.
[40] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.
[41] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–17.
[42] D. Berthelot, T. Schumm, and L. Metz, "BeGAN: Boundary equilibrium generative adversarial networks," 2017, *arXiv:1703.10717*.
[43] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–18.
[44] V. Dumoulin *et al.*, "Adversarially learned inference," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–18.
[45] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "It takes (only) two: Adversarial generator-encoder networks," in *Proc. AAAI Conf. Artifi. Intell.*, 2018, pp. 1250–1257.
[46] T. Nguyen, T. Le, H. Vu, and D. Phung, "Dual discriminator generative adversarial nets," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 2670–2680.
[47] I. Durugkar, I. Gemp, and S. Mahadevan, "Generative multi-adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14.
[48] Q. Hoang, T. D. Nguyen, T. Le, and D. Phung, "MGAN: Training generative adversarial nets with multiple generators," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–24.
[49] A. Ghosh, V. Kulharia, V. P. Namboodiri, P. H. Torr, and P. K. Dokania, "Multi-agent diverse generative adversarial networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8513–8521.
[50] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 469–477.
[51] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.
[52] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 63–79.
[53] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 701–710.
[54] J. Guan, C. Pan, S. Li, and D. Yu, "SRDGAN: Learning the noise prior for super resolution with dual generative adversarial networks," 2019, *arXiv:1903.11821*.
[55] Z. Ding, X.-Y. Liu, M. Yin, W. Liu, and L. Kong, "Tensor super-resolution with generative adversarial nets: A large image generation approach," in *Proc. Int. Workshop Hum. Brain Artif. Intell.*, 2019, pp. 2019–223.
[56] L. Q. Tran, X. Yin, and X. Liu, "Representation learning by rotating your faces," *IEEE Trans. Pattern Analy. Mach. Intell.*, vol. 41, no. 12, pp. 3007–3021, Dec. 2019.
[57] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2439–2448.
[58] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 406–416.
[59] W. Jiang *et al.*, "PSGAN: Pose and expression robust spatial-aware Gan for customizable makeup transfer," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5194–5202.
[60] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "APDrawingGAN: Generating artistic portrait drawings from face photos with hierarchical GANs," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10743–10752.
[61] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.
[62] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Neural photo editing with introspective adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–15.
[63] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2337–2346.
[64] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 702–716.
[65] N. Jetchev, U. Bergmann, and R. Vollgraf, "Texture synthesis with spatial generative adversarial networks," in *Proc. Neural Inf. Process. Syst. Adv. Learn. Workshop*, 2016, pp. 1–11.
[66] U. Bergmann, N. Jetchev, and R. Vollgraf, "Learning texture manifolds with the periodic spatial Gan," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 469–477.
[67] K. Ehsani, R. Mottaghi, and A. Farhadi, "SeGAN: Segmenting and generating the invisible," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6144–6153.
[68] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1222–1230.
[69] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "SOD-MTGAN: Small object detection via multi-task generative adversarial network," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 206–221.
[70] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 613–621.
[71] E. L. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 4414–4423.
[72] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The pose knows: Video forecasting by generating pose futures," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 33321–3341.
[73] T.-C. Wang *et al.*, "Video-to-video synthesis," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 1152–1164.
[74] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing motion and content for video generation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1526–1535.
[75] K. Lin, D. Li, X. He, Z. Zhang, and M.-T. Sun, "Adversarial ranking for language generation," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 3155–3165.
[76] J. Wang *et al.*, "IRGAN: A minimax game for unifying generative and discriminative information retrieval models," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 515–524.
[77] S. Lu, Z. Dou, X. Jun, J.-Y. Nie, and J.-R. Wen, "PSGAN: A minimax game for personalized search with limited and noisy click data," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 555–564.
[78] T. Qiao, J. Zhang, D. Xu, and D. Tao, "MirrorGAN: Learning text-to-image generation by redescription," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1505–1514.
[79] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," in *Proc. Neural Inf. Process. Syst. Constructive Mach. Learn. Workshop*, 2016, pp. 1–6.
[80] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, and A. Aspuru-Guzik, "Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models," 2018, arXiv:1705.10843.
[81] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence generative adversarial nets with policy gradient," in *Proc. AAAI Conf. Artifi. Intell.*, 2017, pp. 2852–2858.
[82] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[83] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Representations*, 2014, pp. 1–14.
[84] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1278–1286.
[85] G. E. Hinton, T. J. Sejnowski, and D. H. Ackley, *Boltzmann Machines: Constraint Satisfaction Networks that Learn*. Pittsburgh, PA, USA: Carnegie-Mellon Univ., 1984.
[86] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cogn. Sci.*, vol. 9, no. 1, pp. 147–169, 1985.

[87] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[88] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 3387–3395.

[89] Y. Bengio, E. Laufer, G. Alain, and J. Yosinski, "Deep generative stochastic networks trainable by backprop," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 226–234.

[90] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *Proc. Neural Inf. Process. Syst.*, 2013, pp. 899–907.

[91] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," 2017, *arXiv:1701.00160*.

[92] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "PixelCNN++: Improving the pixelCNN with discretized logistic mixture likelihood and other modifications," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–10.

[93] B. J. Frey, G. E. Hinton, and P. Dayan, "Does the wake-sleep algorithm produce good density estimators?," in *Proc. Neural Inf. Process. Syst.*, 1996, pp. 661–667.

[94] B. J. Frey, J. F. Brendan, and B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*. Cambridge, MA, USA: MIT Press, 1998.

[95] D. Silver, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[96] K. Eykholt *et al.*, "Robust physical-world attacks on deep learning visual classification," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1625–1634.

[97] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. Int. Conf. Learn. Representations Workshop*, 2017, pp. 1–15.

[98] G. Elsayed *et al.*, "Adversarial examples that fool both computer vision and time-limited humans," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 3910–3920.

[99] X. Jia, X. Wei, X. Cao, and H. Foroosh, "ComDefend: An efficient image compression model to defend adversarial examples," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6084–6092.

[100] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 274–283.

[101] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. SIGKDD Conf. Knowl. Discov. Data Mining*, 2018, pp. 2847–2856.

[102] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9185–9193.

[103] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Representations*, 2014, pp. 1–9.

[104] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–11.

[105] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," in *Proc. IEEE Secur. Privacy Workshops*, 2018, pp. 36–42.

[106] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–17.

[107] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[108] G. Jin, S. Shen, D. Zhang, F. Dai and Y. Zhang, "APE-GAN: Adversarial perturbation elimination with GAN," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 3842–3846.

[109] H. Lee, S. Han, and J. Lee, "Generative adversarial trainer: Defense to adversarial perturbations with GAN," 2017, *arXiv:1705.03387*.

[110] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. Tygar, "Adversarial machine learning," in *Proc. ACM Workshop Secur. Artif. Intell.*, 2011, pp. 43–58.

[111] I. J. Goodfellow, "On distinguishability criteria for estimating generative models," in *Proc. Int. Conf. Learn. Representations Workshop*, 2015, pp. 1–6.

[112] M. Kahng, N. Thorat, D. H. P. Chau, F. B. Viégas, and M. Wattenberg, "GAN lab: Understanding complex deep generative models using interactive visual experimentation," *IEEE Trans. Visualization Comput. Graph.*, vol. 25, no. 1, pp. 1–11, Jan. 2018.

[113] D. Bau *et al.*, "GAN dissection: Visualizing and understanding generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–18.

[114] P. Wu, C. Zheng, and L. Pan, "A unified generative adversarial learning framework for improvement of skip-gram network representation learning methods," *IEEE Trans. Knowl. Data Eng.*, early access, Apr. 30, 2021, doi: 10.1109/TKDE.2021.3076766.

[115] G.-J. Qi, L. Zhang, H. Hu, M. Edraki, J. Wang, and X.-S. Hua, "Global versus localized generative adversarial nets," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1517–1525.

[116] Z. Yu, Z. Zhang, W. Cao, C. Liu, J. P. Chen, and H. San Wong, "GAN-based enhanced deep subspace clustering networks," *IEEE Trans. Knowl. Data Eng.*, early access, Sep., 21, 2021, doi: 10.1109/TKDE.2020.3025301.

[117] M. Kocaoglu, C. Snyder, A. G. Dimakis, and S. Vishwanath, "CausalGAN: Learning causal implicit generative models with adversarial training," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–37.

[118] H. Wang *et al.*, "Learning graph representation with generative adversarial nets," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 8, pp. 3090–3103, Aug. 2021.

[119] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 8559–8570.

[120] Y. Liu *et al.*, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1517–1528, Aug. 2020.

[121] Y. Zhao, Z. Jin, G.-J. Qi, H. Lu, and X.-S. Hua, "An adversarial approach to hard triplet generation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 501–517.

[122] S. Feizi, F. Farnia, T. Ginart, and D. Tse, "Understanding GANs in the LQG setting: Formulation, generalization and stability," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 304–311, May 2020.

[123] F. Farnia and D. Tse, "A convex duality framework for GANs," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 5248–5258.

[124] J. Zhao, J. Li, Y. Cheng, T. Sim, S. Yan, and J. Feng, "Understanding humans in crowded scenes: Deep nested adversarial learning and a new benchmark for multi-human parsing," in *Proc. ACM Multimedia Conf.*, 2018, pp. 792–800.

[125] A. Jahanian, L. Chai, and P. Isola, "On the "steerability" of generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–31.

[126] B. Zhu, J. Jiao, and D. Tse, "Deconstructing generative adversarial networks," *IEEE Trans. Inf. Theory*, vol. 66, no. 11, pp. 7155–7179, Nov. 2020.

[127] K. K. Babu and S. R. Dubey, "CSGAN: Cyclic-synthesized generative adversarial networks for image-to-image transformation," *Expert Syst. Appl.*, vol. 169, pp. 1–12, 2021.

[128] Y. Wu, J. Donahue, D. Balduzzi, K. Simonyan, and T. Lillicrap, "LoGAN: Latent optimisation for generative adversarial networks," 2020, *arXiv:1912.00953*.

[129] T. Kurutach, A. Tamar, G. Yang, S. J. Russell, and P. Abbeel, "Learning plannable representations with causal infoGAN," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 8733–8744.

[130] A. Spurr, E. Aksan, and O. Hilliges, "Guiding infoGAN with semi-supervision," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2017, pp. 119–134.

[131] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4467–4477.

[132] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1–10.

[133] S. Hong, D. Yang, J. Choi, and H. Lee, "Inferring semantic layout for hierarchical text-to-image synthesis," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7986–7994.

[134] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 217–225.

[135] H. Zhang *et al.*, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.

[136] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5077–5086.

[137] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," *Class Project for Stanford CS231N: Convolutional Neural Netw. Vis. Recognit., Winter semester*, vol. 2014, no. 5, 2014, Art. no. 5.

[138] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 2089–2093.

[139] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, "Multichannel attention selection GAN with cascaded semantic guidance for cross-view image translation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2417–2426.

[140] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," 2016, *arXiv:1612.00215*.

[141] B. Dai, S. Fidler, R. Urtasun, and D. Lin, "Towards diverse and natural image descriptions via a conditional GAN," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2970–2979.

[142] S. Yao *et al.*, "3D-aware scene manipulation via inverse graphics," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 1887–1898.

[143] G. G. Chrysos, J. Kossaifi, and S. Zafeiriou, "Robust conditional generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–27.

[144] K. K. Thekumparampil, A. Khetan, Z. Lin, and S. Oh, "Robustness of conditional GANs to noisy labels," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 10271–10282.

[145] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1429–1437.

[146] M. Gong, Y. Xu, C. Li, K. Zhang, and K. Batmanghelich, "Twin auxilary classifiers GAN," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 1328–1337.

[147] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNET: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[148] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional GANs for image editing," in *Proc. Conf. Neural Inf. Process. Syst. Workshop Adversarial Training*, 2016, pp. 1–9.

[149] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2830–2839.

[150] K. Sricharan, R. Bala, M. Shreve, H. Ding, K. Saketh, and J. Sun, "Semi-supervised conditional GANs," 2017, *arXiv:1708.05789*.

[151] T. Miyato and M. Koyama, "cGANS with projection discriminator," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–21.

[152] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[153] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.

[154] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[155] C. Li *et al.*, "Alice: Towards understanding adversarial learning for joint distribution matching," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 5495–5503.

[156] L. C. Tiao, E. V. Bonilla, and F. Ramos, "Cycle-consistent adversarial learning as approximate Bayesian inference," in *Proc. Int. Conf. Mach. Learn. Workshop Theor. Found. Appl. Deep Generative Models*, 2018, pp. 1–17.

[157] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1857–1865.

[158] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2849–2857.

[159] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–17.

[160] A. Yadav, S. Shah, Z. Xu, D. Jacobs, and T. Goldstein, "Stabilizing adversarial nets with prediction methods," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–21.

[161] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.

[162] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs created equal? A large-scale study," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 700–709.

[163] I. Csiszár, P. C. Shields, *et al.*, "Information theory and statistics: A tutorial," *Found. Trends® Commun. Inf. Theory*, vol. 1, no. 4, pp. 417–528, 2004.

[164] D. J. Im, H. Ma, G. Taylor, and K. Branson, "Quantitatively evaluating Gans with divergences proposed for training," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–30.

[165] M. Uehara, I. Sato, M. Suzuki, K. Nakayama, and Y. Matsuo, "Generative adversarial nets from a density ratio estimation perspective," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–16.

[166] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, "Hilbert space embeddings and metrics on probability measures," *J. Mach. Learn. Res.*, vol. 11, pp. 1517–1561, 2010.

[167] A. Uppal, S. Singh, and B. Poczos, "Nonparametric density estimation & convergence of GANs under besov IPM losses," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 9086–9097.

[168] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012. [Online]. Available: https://jmlr.org/papers/volume13/gretton12a/gretton12a.pdf

[169] K. M. Borgwardt, A. Gretton, M. J. Rasch, H. P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[170] W. Wang, Y. Sun, and S. Halgamuge, "Improving MMD-GAN training with repulsive loss function," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–24.

[171] M. Arbel, D. Sutherland, M. Bińkowski, and A. Gretton, "On gradient regularizers for MMD GANs," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 6700–6710.

[172] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," 2021, *arXiv:1801.01401*.

[173] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos, "MMD GAN: Towards deeper understanding of moment matching network," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 2203–2213.

[174] Y. Mroueh, C.-L. Li, T. Sercu, A. Raj, and Y. Cheng, "Sobolev GAN," in *Proc. Int. Conf. Learn. Representations*, 2018.

[175] D. J. Sutherland *et al.*, "Generative models and model criticism via optimized maximum mean discrepancy," 2021, *arXiv:1611.04488*.

[176] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via maximum mean discrepancy optimization," in *Proc. Conf. Uncertainty Artif. Intell.*, 2015, pp. 258–267.

[177] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1718–1727.

[178] C. Villani, *Optimal Transport: Old and New*, Berlin, Germany: Springer, 2008.

[179] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, "Stabilizing training of generative adversarial networks through regularization," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 2018–2028.

[180] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for GANs do actually converge?," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3481–3490.

[181] W. Fedus, M. Rosca, B. Lakshminarayanan, A. M. Dai, S. Mohamed, and I. Goodfellow, "Many paths to equilibrium: GANs do not need to decrease a divergence at every step," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–18.

[182] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, "On convergence and stability of GANs," 2017, *arXiv:1705.07215*.

[183] J. Wu, Z. Huang, J. Thoma, D. Acharya, and L. Van Gool, "Wasserstein divergence for GANs," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 653–668.

[184] M. G. Bellemare *et al.*, "The cramer distance as a solution to biased wasserstein gradients," 2017, *arXiv:1705.10743*.

[185] H. Petzka, A. Fischer, and D. Lukovnicov, "On the regularization of wasserstein GANs," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–24.

[186] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," in *Proc. Interspeech*, 2017, pp. 3364–3368.

[187] J. Adler and S. Lunz, "Banach wasserstein GAN," in *Proc. Neural Info. Process. Syst.*, 2018, pp. 6754–6763.

[188] Y.-S. Chen, Y.-C. Wang, M.-H. Kao, and Y.-Y. Chuang, "Deep photo enhancer: Unpaired learning for image enhancement from photographs with GANs," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6306–6314.

[189] S. Athey, G. Imbens, J. Metzger, and E. Munro, "Using wasserstein generative adversarial networks for the design of monte carlo simulations," *J. Econometrics*, 2021.

[190] A. Krizhevsky, "Learning multiple layers of features from tiny images," State College, PA, USA, Tech. Rep., 2009.

[191] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[192] J. Susskind, A. Anderson, and G. E. Hinton, "The toronto face dataset," Univ. Toronto, ON, Canada, UTML, Tech. Rep. UTML TR, 2010.

[193] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. Int. Conf. Learn. Representations Workshop*, 2015, pp. 1–14.

[194] A. A. Rusu *et al.*, "Progressive neural networks," 2016, *arXiv:1606.04671*.

[195] T. Xu *et al.*, "AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1316–1324.

[196] T. R. Shaham, T. Dekel, and T. Michaeli, "SinGAN: Learning a generative model from a single natural image," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 4570–4580.

[197] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[198] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "VEEGAN: Reducing mode collapse in GANs using implicit variational learning," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 3308–3318.

[199] D. Bau *et al.*, "Seeing what a GAN cannot generate," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 4502–4511.

[200] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (GANs)," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 224–232.

[201] L. Mescheder, S. Nowozin, and A. Geiger, "The numerics of GANs," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1825–1835.

[202] Z. Lin, A. Khetan, G. Fanti, and S. Oh, "PacGAN: The power of two samples in generative adversarial networks," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 1498–1507.

[203] S. Arora, A. Risteski, and Y. Zhang, "Do GANs learn the distribution? Some theory and empirics," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–16.

[204] Y. Bai, T. Ma, and A. Risteski, "Approximability of discriminators implies diversity in GANs," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–45.

[205] T. Liang, "How well generative adversarial networks learn distributions," 2020, *arXiv:1811.03179*.

[206] S. Singh, A. Uppal, B. Li, C.-L. Li, M. Zaheer, and B. Póczos, "Nonparametric density estimation with adversarial losses," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 10246–10257.

[207] S. Mohamed and B. Lakshminarayanan, "Learning in implicit generative models," 2017, *arXiv:1610.03483*.

[208] G. Gidel, H. Berard, G. Vignoud, P. Vincent, and S. Lacoste-Julien, "A variational inequality perspective on generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–38.

[209] M. Sanjabi, J. Ba, M. Razaviyayn, and J. D. Lee, "On the convergence and robustness of training GANs with regularized optimal transport," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 7091–7101.

[210] V. Nagarajan, C. Raffel, and I. J. Goodfellow, "Theoretical insights into memorization in GANs," in *Proc. Neural Inf. Process. Syst. Workshop*, 2018, pp. 1–10.

[211] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, "The 2018 PIRM challenge on perceptual image super-resolution," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 334–355.

[212] X. Yu and F. Porikli, "Ultra-resolving face images by discriminative generative networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 318–333.

[213] H. Zhu, A. Zheng, H. Huang, and R. He, "Arbitrary talking face generation via attentional audio-visual coherence learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 2362–2368.

[214] H. Huang, R. He, Z. Sun, and T. Tan, "Wavelet domain generative adversarial network for multi-scale face hallucination," *Int. J. Comput. Vis.*, vol. 127, no. 6–7, pp. 763–784, 2019.

[215] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, "Amortised MAP inference for image super-resolution," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–17.

[216] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[217] X. Wang, K. Yu, C. Dong, and C. Change Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 606–615.

[218] W. Zhang, Y. Liu, C. Dong, and Y. Qiao, "RankSRGAN: Generative adversarial networks with ranker for image super-resolution," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 3096–3105.

[219] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning GAN for pose-invariant face recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1415–1424.

[220] J. Cao, Y. Hu, H. Zhang, R. He, and Z. Sun, "Learning a high fidelity pose invariant model for high-resolution face frontalization," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 2867–2877.

[221] A. Siarohin, E. Sangineto, S. Lathuilière, and N. Sebe, "Deformable GANs for pose-based human image generation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3408–3416.

[222] C. Wang, C. Wang, C. Xu, and D. Tao, "Tag disentangled generative adversarial networks for object image re-rendering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2901–2907.

[223] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras, "Neural face editing with intrinsic image disentangling," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5541–5550.

[224] H. Chang, J. Lu, F. Yu, and A. Finkelstein, "Pairedcyclegan: Asymmetric style transfer for applying and removing makeup," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 40–48.

[225] B. Dolhansky and C. Canton Ferrer, "Eye in-painting with exemplar generative adversarial networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7902–7911.

[226] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "Ganimation: Anatomically-aware facial animation from a single image," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 818–833.

[227] C. Donahue, Z. C. Lipton, A. Balsubramani, and J. McAuley, "Semantically decomposing the latent spaces of generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–19.

[228] A. Duarte *et al.*, "Wav2Pix: Speech-conditioned face generation using generative adversarial networks," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 8633–8637.

[229] B. Gecer, S. Ploumpis, I. Kotsia, and S. Zafeiriou, "GANFIT: Generative adversarial network fitting for high fidelity 3D face reconstruction," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1155–1164.

[230] Z. Shu, M. Sahasrabudhe, R. Alp Guler, D. Samaras, N. Paragios, and I. Kokkinos, "Deforming autoencoders: Unsupervised disentangling of shape and appearance," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 650–665.

[231] C. Fu, X. Wu, Y. Hu, H. Huang, and R. He, "Dual variational generation for low shot heterogeneous face recognition," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 2670–2679.

[232] J. Cao, Y. Hu, B. Yu, R. He, and Z. Sun, "3D aided duet GANs for multi-view face image synthesis," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 8, pp. 2028–2042, Aug. 2019.

[233] Y. Liu, Q. Li, and Z. Sun, "Attribute-aware face aging with wavelet-based generative adversarial networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11877–118866.

[234] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: Fine-grained image generation through asymmetric training," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2745–2754.

[235] H. Dong, S. Yu, C. Wu, and Y. Guo, "Semantic image synthesis via adversarial learning," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 5706–5714.

[236] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 82–90.

[237] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, "Generating images with recurrent adversarial networks," 2016, *arXiv:1602.05110*.

[238] J. Yang, A. Kannan, D. Batra, and D. Parikh, "LR-GAN: Layered recursive generative adversarial networks for image generation," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–21.

[239] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-RCNN: Hard positive generation via adversary for object detection," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2606–2615.

[240] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–22.

[241] E. Santana and G. Hotz, "Learning a driving simulator," 2016, *arXiv:1608.01230*.

[242] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 5932–5941.

[243] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.

[244] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion GAN for future-flow embedded video prediction," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 1744–1752.

[245] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, "Recycle-GAN: Unsupervised video retargeting," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 119–135.

[246] X. Liang, H. Zhang, L. Lin, and E. Xing, "Generative semantic manipulation with mask-contrasting GAN," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 558–573.

[247] Y. Chen, Y.-K. Lai, and Y.-J. Liu, "CartoonGAN: Generative adversarial networks for photo cartoonization," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9465–9474.

[248] R. Villegas, J. Yang, D. Ceylan, and H. Lee, "Neural kinematic networks for unsupervised motion retargetting," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8639–8648.

[249] S. Zhou, T. Xiao, Y. Yang, D. Feng, Q. He, and W. He, "GeneGAN: Learning object transfiguration and attribute subspace from unpaired data," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–13.

[250] H. Wu, S. Zheng, J. Zhang, and K. Huang, "GP-GAN: Towards realistic high-resolution image blending," in *Proc. ACM Int. Conf. Multimedia*, 2019, pp. 2487–2495.

[251] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 5688–5696.

[252] J. Pan *et al.*, "SalGAN: Visual saliency prediction with generative adversarial networks," 2018, *arXiv:1701.01081*.

[253] Y. Song *et al.*, "Vital: Visual tracking via adversarial learning," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8990–8999.

[254] Y. Han, P. Zhang, W. Huang, Y. Zha, G. D. Cooper, and Y. Zhang, "Robust visual tracking based on adversarial unlabeled instance generation with label smoothing loss regularization," *Pattern Recognit.*, vol. 97, pp. 1–15, 2020.

[255] D. Engin, A. Genç, and H. Kemal Ekenel, "Cycle-dehaze: Enhanced cycleGAN for single image dehazing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 825–833.

[256] X. Yang, Z. Xu, and J. Luo, "Towards perceptual image dehazing by physics-based disentanglement and adversarial training," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7485–7492.

[257] W. Liu, X. Hou, J. Duan, and G. Qiu, "End-to-end single image fog removal using enhanced cycle consistent adversarial networks," *IEEE Trans. Image Process.*, vol. 29, pp. 7819–7833, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9139368

[258] S. Lutz, K. Amplianitis, and A. Smolic, "AlphaGAN: Generative adversarial networks for natural image matting," in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 1–17.

[259] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5485–5493.

[260] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5505–5514.

[261] X. Liu, Y. Wang, and Q. Liu, "PsGAN: A generative adversarial network for remote sensing image pan-sharpening," in *Proc. IEEE Int. Conf. Image Process.*, 2018, pp. 873–877.

[262] S. Zhao, J. Cui, Y. Sheng, Y. Dong, X. Liang, E. I. Chang, and Y. Xu, "Large scale image completion via co-modulated generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–25.

[263] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, 2017.

[264] F. Liu, L. Jiao, and X. Tang, "Task-oriented GAN for polSAR image classification and clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2707–2719, Sep. 2019.

[265] A. Creswell and A. A. Bharath, "Adversarial training for sketch retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 798–809.

[266] M. Zhang, K. Teck Ma, J. Hwee Lim, Q. Zhao, and J. Feng, "Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4372–4381.

[267] M. Zhang, K. T. Ma, J. Lim, Q. Zhao, and J. Feng, "Anticipating where people will look using adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1783–1796, Aug. 2019.

[268] F. Fang, J. Yamagishi, I. Echizen, and J. Lorenzo-Trueba, "High-quality nonparallel voice conversion based on cycle-consistent adversarial network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5279–5283.

[269] T. Kaneko and H. Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," 2017, *arXiv:1711.11293*.

[270] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional GANs," 2017, *arXiv:1706.02633*.

[271] K. G. Hartmann, R. T. Schirrmeister, and T. Ball, "Eeg-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals," 2018, *arXiv:1806.01875*.

[272] C. Donahue, J. McAuley, and M. Puckette, "Synthesizing audio with generative adversarial networks," 2018, *arXiv:1802.04208*.

[273] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. Int. Conf. Artif. Neural Netw.*, 2019, pp. 703–716.

[274] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2157–2169.

[275] Y. Zhang, Z. Gan, and L. Carin, "Generating text via adversarial training," in *Proc. Conf. Neural Inf. Process. Syst. Workshop Adv. Training*, 2016, pp. 1–6.

[276] W. Fedus, I. Goodfellow, and A. M. Dai, "MaskGAN: Better text generation via filling in the _," in *Proc. Int. Confe. Learn. Representations*, 2018, pp. 1–16.

[277] S. Yang, J. Liu, W. Wang, and Z. Guo, "TET-GAN: Text effects transfer via stylization and destylization," in *Proc. AAAI Conf. Artif. Intelli.*, 2019, pp. 1238–1245.

[278] L. Cai and W. Y. Wang, "KBGAN: Adversarial learning for knowledge graph embeddings," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2018, pp. 1–10.

[279] X. Wang, W. Chen, Y.-F. Wang, and W. Y. Wang, "No metrics are perfect: Adversarial reward learning for visual storytelling," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1–15.

[280] P. Qin, W. Xu, and W. Y. Wang, "Dsgan: generative adversarial training for distant supervision relation extraction," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1–10.

[281] C. d. M. d'Autume, M. Rosca, J. Rae, and S. Mohamed, "Training language GANs from scratch," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 4302–4313.

[282] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Liwicki, and M. Z. Afzal, "TAC-GAN-text conditioned auxiliary classifier generative adversarial network," 2017, *arXiv:1703.06412*.

[283] T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, and M. Sun, "Show, adapt and tell: Adversarial training of cross-domain image captioner," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 521–530.

[284] R. Shetty, M. Rohrbach, L. Anne Hendricks, M. Fritz, and B. Schiele, "Speaking the same language: Matching machine to human captions by adversarial training," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 4135–4144.

[285] S. Rao and H. Daum é III, “Answer-based adversarial training for generating clarification questions,” in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 1–14.
[286] X. Yang *et al.*, “Adversarial training for community question answer selection based on multi-scale matching,” in *Proc. AAAI Conf. Artifi. Intell.*, 2019, pp. 395–402.
[287] B. Liu, J. Fu, M. P. Kato, and M. Yoshikawa, “Beyond narrative description: Generating poetry from images by multi-adversarial training,” in *Proc. ACM Multimedia Conf.*, 2018, pp. 783–791.
[288] Y. Luo, H. Zhang, Y. Wen, and X. Zhang, “ResumeGAN: An optimized deep representation learning framework for talent-job fit via adversarial learning,” in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 1101–1110.
[289] C. Garbacea, S. Carton, S. Yan, and Q. Mei, “Judge the judges: A large-scale evaluation study of neural language models for online review generation,” in *Proc. Conf. Empirical Methods Natural Lang. Process. Int. Joint Conf.*, 2019, pp. 3966–3979.
[290] H. Aghakhani, A. Machiry, S. Nilizadeh, C. Kruegel, and G. Vigna, “Detecting deceptive reviews using generative adversarial networks,” in *Proc. IEEE Secur. Privacy Workshops*, 2018, pp. 89–95.
[291] K. Takuhiro, K. Hirokazu, H. Nobukatsu, I. Yusuke, H. Kaoru, and K. Kunio, “Generative adversarial network-based postfiltering for statistical parametric speech synthesis,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 4910–4914.
[292] Y. Saito, S. Takamichi, and H. Saruwatari, “Statistical parametric speech synthesis incorporating generative adversarial networks,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 1, pp. 84–96, Jan. 2018.
[293] C. Donahue, J. McAuley, and M. Puckette, “Adversarial audio synthesis,” in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–16.
[294] S. Pascual, A. Bonafonte, and J. Serra, “SeGAN: Speech enhancement generative adversarial network,” in *Proc. Interspeech*, 2017, pp. 3642–3646.
[295] C. Donahue, B. Li, and R. Prabhavalkar, “Exploring speech enhancement with generative adversarial networks for robust speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5024–5028.
[296] N. Killoran, L. J. Lee, A. Delong, D. Duvenaud, and B. J. Frey, “Generating and designing dna with deep generative models,” in *Proc. Conf. Neural Inf. Process. Syst. Comput. Biol. Workshop*, 2017, pp. 1–19.
[297] A. Gupta and J. Zou, “Feedback GAN for DNA optimizes protein functions,” *Nat. Mach. Intell.*, vol. 1, no. 2, pp. 105–111, 2019.
[298] M. Benhenda, “Chemgan challenge for drug discovery: Can AI reproduce natural chemical diversity?,” 2017, *arXiv:1708.08227*.
[299] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, “Generating multi-label discrete patient records using generative adversarial networks,” in *Proc. Mach. Learn. Healthcare*, 2017, pp. 1–20.
[300] W. Dai *et al.*, “SCAN: Structure correcting adversarial network for organ segmentation in chest X-rays,” in *Proc. Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support*, 2018, pp. 1–10.
[301] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” in *Proc. Int. Conf. Inf. Process. Med. Imaging*, 2017, pp. 146–157.
[302] J. M. Wolterink, A. M. Dinkla, M. H. Savenije, P. R. Seevinck, C. A. van den Berg, and I. Išgum, “Deep MR to CT synthesis using unpaired data,” in *Proc. Int. Workshop Simulation Synth. Med. Imaging*, 2017, pp. 14–23.
[303] T. M. Quan, T. Nguyen-Duc, and W.-K. Jeong, “Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss,” *IEEE Trans. Med. Imaging*, vol. 37, no. 6, pp. 1488–1497, Jun. 2018.
[304] M. Mardani *et al.*, “Deep generative adversarial neural networks for compressive sensing mri,” *IEEE Trans. Med. Imaging*, vol. 38, no. 1, pp. 167–179, Jan. 2019.
[305] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang, “SeGAN: Adversarial network with multi-scale l_1 loss for medical image segmentation,” *Neuroinformatics*, vol. 16, no. 3–4, pp. 383–392, 2018.
[306] Q. Yang *et al.*, “Low-dose ct image denoising using a generative adversarial network with Wasserstein distance and perceptual loss,” *IEEE Trans. Med. Imaging*, vol. 37, no. 6, pp. 1348–1357, Jun. 2018.
[307] G. St-Yves and T. Naselaris, “Generative adversarial networks conditioned on brain activity reconstruct seen images,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern*, 2018, pp. 1054–1061.
[308] B. Tian, Y. Zhang, X. Chen, C. Xing, and C. Li, “DRGAN: A gan-based framework for doctor recommendation in chinese on-line QA communities,” in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2019, pp. 444–447.
[309] Z. Zheng, L. Zheng, and Y. Yang, “Unlabeled samples generated by GAN improve the person re-identification baseline in vitro,” in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 3754–3762.
[310] B. Chang, Q. Zhang, S. Pan, and L. Meng, “Generating handwritten chinese characters using cyclegan,” in *Proc. IEEE Winter Confe. Appl. Comput. Vis.*, 2018, pp. 199–207.
[311] L. Sixt, B. Wild, and T. Landgraf, “RenderGAN: Generating realistic labeled data,” *Front. Robot. AI*, vol. 5, pp. 1–9, 2018.
[312] D. Xu, S. Yuan, L. Zhang, and X. Wu, “FairGAN: Fairness-aware generative adversarial networks,” in *Proc. IEEE Int. Conf. Big Data*, 2018, pp. 570–575.
[313] M.-C. Lee, B. Gao, and R. Zhang, “Rare query expansion through generative adversarial networks in search advertising,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 500–508.
[314] D. Xu, Y. Wu, S. Yuan, L. Zhang, and X. Wu, “Achieving causal fairness through generative adversarial networks,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 1452–1458.
[315] M. O. Turkoglu, W. Thong, L. Spreeuwers, and B. Kicanaoglu, “A layer-based sequential framework for scene generation with gans,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8901–8908.
[316] A. El-Nouby *et al.*, “Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction,” in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 10303–10311.
[317] N. Ratzlaff and L. Fuxin, “HyperGAN: A generative model for diverse, performant neural networks,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5361–5369.
[318] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, “GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification,” *Neurocomputing*, vol. 321, pp. 321–331, 2018.
[319] Q. Wang, H. Yin, H. Wang, Q. V. H. Nguyen, Z. Huang, and L. Cui, “Enhancing collaborative filtering with generative augmentation,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 548–556.
[320] Y. Zhang, Y. Fu, P. Wang, X. Li, and Y. Zheng, “Unifying inter-region autocorrelation and intra-region structures for spatial embedding via collective adversarial learning,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1700–1708.
[321] H. Gao, J. Pei, and H. Huang, “ProGAN: Network embedding via proximity generative adversarial network,” in *Proc. ACM SIGKDD Inte. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1308–1316.
[322] B. Hu, Y. Fang, and C. Shi, “Adversarial learning on heterogeneous information networks,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 120–129.
[323] P. Wang, Y. Fu, H. Xiong, and X. Li, “Adversarial substructured representation learning for mobile user profiling,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 130–138.
[324] W. Hu and Y. Tan, “Generating adversarial malware examples for black-box attacks based on GAN,” 2017, *arXiv:1702.05983*.
[325] C. Chu, A. Zhmoginov, and M. Sandler, “CycleGAN: A master of steganography,” in *Proc. Confe. Neural Inf. Process. Syst. Workshop Mach. Deception*, 2017, pp. 1–6.
[326] D. Volkhonskiy, I. Nazarov, B. Borisenko, and E. Burnaev, “Steganographic generative adversarial networks,” in *Proc. Confe. Neural Inf. Process. Syst. Workshop Adv. Training*, 2016, pp. 1–15.
[327] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, “SSGAN: Secure steganography based on generative adversarial networks,” in *Proc. Pacific Rim Conf. Multimedia*, 2017, pp. 534–544.
[328] J. Hayes and G. Danezis, “Generating steganographic images via adversarial training,” in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1954–1963.
[329] M. Abadi and D. G. Andersen, “Learning to protect communications with adversarial neural cryptography,” 2016, *arXiv:1610.06918*.

[330] A. N. Gomez, S. Huang, I. Zhang, B. M. Li, M. Osama, and L. Kaiser, “Unsupervised cipher cracking using discrete GANs,” in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–15.

[331] B. K. Beaulieu-Jones *et al.*, “Privacy-preserving generative deep neural networks support clinical data sharing,” *Circulation: Cardiovasc. Quality Outcomes*, vol. 12, no. 7, pp. 1–10, 2019.

[332] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social GAN: Socially acceptable trajectories with generative adversarial networks,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2255–2264.

[333] H. Shu *et al.*, “Co-evolutionary compression for unpaired image translation,” in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 3234–3243.

[334] S. Lin *et al.*, “Towards optimal structured CNN pruning via generative adversarial learning,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2790–2799.

Jie Gui (Senior Member,IEEE) received the BS degree in computer science from Hohai University, Nanjing, China, in 2004, the MS degree in computer applied technology from the Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei, China, in 2007, and the PhD degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2010. He is currently a professor with the School of Cyber Science and Engineering, Southeast University. His research interests include machine learning, pattern recognition, and image processing. He is currently the associate editor for *Neurocomputing*. He has authored or coauthored more than 60 papers in international journals and conferences, including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Cybernetics*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE TSMCS, KDD, AAAI, and ACM MM. He is the area chair, senior PC member, or PC member of many conferences, including NeurIPS and ICML. He is a senior member of the ACM and a CCF distinguished member.

Zhenan Sun (Senior Member, IEEE) received the BE degree in industrial automation from the Dalian University of Technology, Dalian, China, in 1999, the MS degree in system engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2002, and the PhD degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2006. Since 2006, he has been a faculty member with the National Laboratory of Pattern Recognition, CASIA, where he is currently a professor with the Center for Research on Intelligent Perception and Computing.

Yonggang Wen (Fellow, IEEE) received the PhD degree in electrical engineering and computer science (with a minor in Western literature) from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008. He is currently a professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has been with Cisco, San Jose, CA, USA, where he led product development in a content delivery network, which had a revenue impact of $3 billion globally. His work in multi-screen cloud social TV has been featured by global media (more than 1600 news articles from more than 29 countries). He has authored or coauthored more than 140 papers in top journals and prestigious conferences.

Dacheng Tao (Fellow, IEEE) is currently a professor of computer science and ARC laureate fellow with the School of Computer Science and the Faculty of Engineering, and the inaugural director of the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney. His research interest include artificial intelligence have been expounded in one monograph and more than 200 publications in prestigious journals and at prominent conferences, including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IJCV, JMLR, AAAI, IJCAI, NIPS, ICML, CVPR, ICCV, ECCV, ICDM, and KDD, with several best paper awards. He was the Recipient of the 2018 IEEE ICDM Research Contributions Award and the 2015 Australian Scopus-Eureka prize. He is a fellow of the ACM and the Australian Academy of Science.

Jieping Ye (Fellow, IEEE) recevied the PhD degree in computer science and engineering from the University of Minnesota in 2005. He is currently a VP of Beike, China, and also a Professor with the University of Michigan, Ann Arbor, MI, USA. His research interests include Big Data, machine learning, and data mining with applications in transportation and biomedicine. He was a senior program committee or the area chair or the program committee vice chair of many conferences, including NIPS, ICML, KDD, IJCAI, ICDM, and SDM. He was an associate editor for *Data Mining and Knowledge Discovery* and *IEEE Transactions on Knowledge and Data Engineering*. He was the Recipient of the best paper awards at ICML and KDD and NSF CAREER Award in 2010.