



# Games of GANs: game-theoretical models for generative adversarial networks

Monireh Mohebbi Moghaddam<sup>1,3</sup> · Bahar Boroomand<sup>1,4</sup> · Mohammad Jalali<sup>1</sup> · Arman Zareian<sup>1</sup> · Alireza Daeijavad<sup>1,5</sup> · Mohammad Hossein Manshaei<sup>1,2</sup> · Marwan Krunz<sup>2</sup>

© The Author(s), under exclusive licence to Springer Nature B.V. 2023

## Abstract

Generative Adversarial Networks (GANs) have recently attracted considerable attention in the AI community due to their ability to generate high-quality data of significant statistical resemblance to real data. Fundamentally, GAN is a game between two neural networks trained in an adversarial manner to reach a zero-sum Nash equilibrium profile. Despite the improvement accomplished in GANs in the last few years, several issues remain to be solved. This paper reviews the literature on the game-theoretic aspects of GANs and addresses how game theory models can address specific challenges of generative models and improve the GAN's performance. We first present some preliminaries, including the basic GAN model and some game theory background. We then present a taxonomy to classify state-of-the-art solutions into three main categories: modified game models, modified architectures, and modified learning methods. The classification is based on modifications made to the basic GAN model by proposed game-theoretic approaches in the literature. We then explore the objectives of each category and discuss recent works in each class. Finally, we discuss the remaining challenges in this field and present future research directions.

**Keywords** Generative adversarial network (GAN) · Game theory · Multi-agent systems · Deep generative models · Deep learning

## 1 Introduction

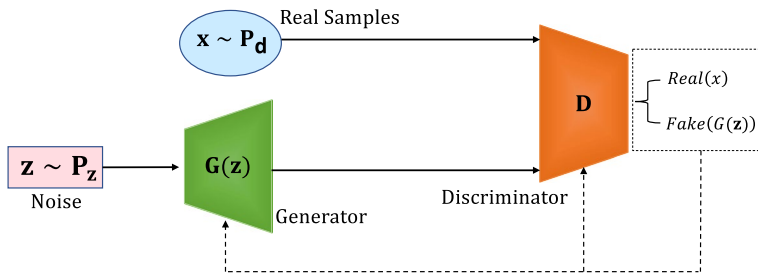
Generative Adversarial Networks (GANs) represent a class of generative models that was originally proposed by Goodfellow et al. (2014b). These models have received wide attention in recent years due to their potential to model high-dimensional complex real-world data (Hong et al. 2019). As generative models, GANs do not minimize a single training criterion. They are used to estimate the real data probability distribution. A

---

Bahar Boroomand, Mohammad Jalali, and Arman Zareian have contributed equally to this work.

✉ Mohammad Hossein Manshaei  
[manshaei@arizona.edu](mailto:manshaei@arizona.edu); [manshaei@iut.ac.ir](mailto:manshaei@iut.ac.ir)

Extended author information available on the last page of the article

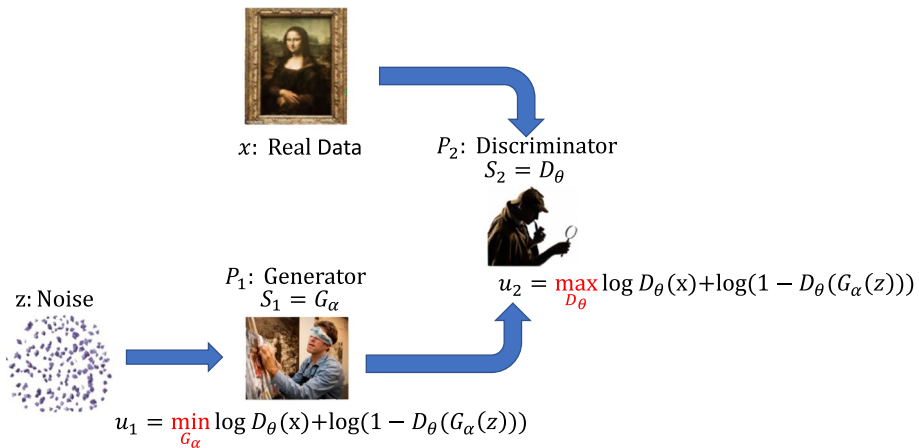


**Fig. 1** GAN's basic architecture (Wang et al. 2017)

GAN usually comprises two neural networks: a discriminator and a generator, which are trained simultaneously via an adversarial learning technique. Such a GAN is more powerful in both feature learning and representation (Fedus et al. 2017). The discriminator attempts to differentiate between real data samples and fake samples made by the generator, while the generator tries to create realistic samples that cannot be distinguished by the discriminator (Li et al. 2018), as shown in Fig. 1. In particular, many GAN models do not rely on any assumptions about the distribution of the real data they are trying to mimic. They can generate infinitely realistic new samples from latent space (Hong et al. 2019). This feature enables GANs to be successfully applied in various applications, ranging from image synthesis, computer vision, video and animation generation to cybersecurity (Alqahtani et al. 2021).

The core idea of GANs is inspired by a two-player zero-sum minimax game between the discriminator and the generator. In such a game, the total utilities of two players are zero, and each player's gain or loss exactly balances by the loss or gain of the other player. GANs are designed to reach a Nash equilibrium at which each player cannot increase its gain without reducing the other player's gain (Goodfellow et al. 2014b; Wang et al. 2017). To better comprehend this concept, we can view this two-player game as a student-teacher game, where the generator plays the role of a student, trying to draw a painting, and the discriminator plays as a teacher, trying to teach the student how to draw. To succeed in this game, the student must learn to make indistinguishable paintings from genuine artworks, and the teacher must learn to discriminate between real and fake paintings. In order to play at the Nash equilibrium, the two players need to continuously optimize their best response strategies to enhance the generating and the discriminating abilities, respectively. Figure 2 shows this game, considering players, strategy sets, and the payoffs. Players ( $P_1$  and  $P_2$ ) are like the generator and the discriminator in a GAN setup shown in Fig. 1. The strategy sets of players ( $S_1$  and  $S_2$ ) is making indistinguishable paintings from genuine arts by the student and discriminate between real and fake paintings by the teacher. These strategies correspond to the weights of neural networks in GAN, i.e., choosing best weights  $\theta$  and  $\alpha$  in discriminator ( $D_\theta$ ) and generator ( $G_\alpha$ ) neural networks, respectively. Players attempt to maximize their payoff ( $u_1$  and  $u_2$  shown in Fig. 2) to reach a Nash equilibrium point.

Despite the significant success of GANs in many domains, applying them to certain real-world problems has been hindered by various challenges. The most significant of these problems is that GANs are hard to train and suffer from instability problems, such as mode collapse, non-convergence, and vanishing gradients. A GAN needs to converge to the Nash equilibrium during the training process, but such convergence has been shown to be challenging (Wang et al. 2019; Wiatrak and Albrecht 2019).



**Fig. 2** Example of student-teacher game as a GAN: players, strategy sets, and payoffs

Since their introduction in 2014, GANs have been widely studied, and numerous methods have been proposed to address their challenges. However, to synthesize high-quality data using GANs, it is necessary to improve their underlying theory, which has been a major hindrance to developing GANs (Cao et al. 2018). As the basic principle of GANs is based on game theory, exploiting the game-theoretical techniques became one of the most discussed topics and attracted research efforts in recent years.

## 1.1 Motivation and contribution

As we mentioned in Sect. 1, GANs are essentially a competition between a generator and a discriminator which are trained iteratively in an adversarial learning manner. Given the way the GANs are trained, the GAN frameworks are naturally modeled and analyzed as a two-players zero-sum game and the Nash equilibrium of the game is a saddle point where the generator has captured the distribution of real samples. Based on the basic relation between these two concepts, recently, researchers have made good progress in improving GAN results and mitigating challenges in this field by utilizing the concepts of the game theory. This primarily motivated us to review these researches. Another important motivation for focusing on game-theoretical progress in GANs is the absence of a comprehensive survey that addresses the recent advances in the GANs from a game-theoretic perspective. Other published surveys (e.g. Goodfellow 2016; Wang et al. 2017; Creswell et al. 2018; Hitawala 2018; Gonog and Zhou 2019; Hong et al. 2019; Bissoto et al. 2019; Zhang et al. 2018a; Pan et al. 2019; Kumar and Jayagopal 2021; Salehi et al. 2020; Ghosh et al. 2020; Saxena and Cao 2020; Gui et al. 2020; Jabbar et al. 2020) considered other aspects of GANs, as explained later. To the best of our knowledge, this work is the first to explore the GAN advancement from a game-theoretic standpoint.

Our survey first introduces some background and key concepts in this field. Then we classify recently proposed *games of GANs* models into three major categories: (i) modified game models, (ii) modified architectures in terms of the number of agents, and (iii) modified learning methods. We then classify each group into several subcategories and review the main contributions of each work. We also address some existing problems in the discussed literature and highlight possible future research directions.

## 1.2 Paper structure and organization

The rest of this paper is organized as follows. Section 2 presents some background on game theory and GANs including the basic ideas, learning method, and challenges. In Sect. 3 we have a glimpse of the other surveys conducted in the field of GANs. We provide our proposed taxonomy in Sect. 4 and review the research models in each category in this section. Section 5 is devoted to discussion and conclusions.

## 2 Background and preliminaries

We start with an overview of game theory and then move toward GANs. Table 1 lists the acronyms and their definitions used throughout the paper.

### 2.1 Game theory

Game theory aims to model the interactions between several decision-makers. These interactions are called a “game” and decision-makers are called “players”. In each round of the game, players take actions from a set of actions called strategy set. It is usually assumed that players are rational, which means that each one of them tries to choose an action that maximizes its payoff with respect to the other players’ actions. So, each agent should have a belief about the other players (Osborne 2004).

Games can be categorized in different ways. We can classify games based on the timing of the players’ actions into two categories: simultaneous and sequential games. In simultaneous games, all players choose their actions concurrently, or, at least without knowing the actions chosen by other players. On the contrary, in sequential games one player decides on its course of action before the others do. To ensure that the discrepancy in time has no strategic impact, the other players must be aware of the first player’s decision. Every time a new step is taken, both players will be fully aware of how the previous ones went (Osborne 2004).

One of the simplest games are two-player constant-sum games, which are games where the sum of the two players’ utilities is equal to this amount in all other states. When this amount equals zero, the game is called *zero-sum game* (Shoham and Leyton-Brown 2008).

Several solutions have been introduced for analyzing games and finding their points of equilibrium. One type of equilibrium is “Nash equilibrium (NE)” in which no player can increase its payoff by changing its strategy unilaterally. In the other words, Nash equilibrium is a state where nobody regrets his/her choice given others’ strategies and with respect to his/her payoff (Shoham and Leyton-Brown 2008). NE in sequential games is known as Subgame Perfect (Nash) Equilibrium (SPNE), which is a Nash equilibrium with the characteristic that all players play their best moves after the history of each game (Brocas et al. 2018). In the situation where the players assign a probability distribution to the strategy sets instead of choosing one strategy, the Nash equilibrium is called *mixed Nash equilibrium* (Shoham and Leyton-Brown 2008). Although Nash proved every finite game has a mixed NE, pure NE does not exist in all games (Osborne 2004).

Considering zero-sum games, we can define another solution concept named maximin or minimax strategy. In the maximin strategy, the decision maker maximizes its worst-case payoff, which happens when all other players cause as much harm as they can to the

**Table 1** Acronyms and corresponding full names appearing in the paper

Acronym	Full name
AE	Auto-encoder
ASRFB	Averaging over decision variables
CS-GAN	Cyclic-synthesized GAN
DDL	Discriminator discrepancy loss
DDPG	Deep deterministic policy gradient
FedGAN	Federated GAN
FID	Fréchet inception distance
GAN	Generative adversarial network
IID	Independent and identically distributed
IRL	Inverse reinforcement learning
IS	Inception score
JSD	Jensen-Shannon divergence
KL	Kullback-Leibler
L-GAN	Latent-space GAN
MAD-GAN	Multi-agent diverse GAN
MADGAN	Multiagent distributed GAN
MolGAN	Molecular GAN
MPM GAN	Message passing multi-agent GAN
MS	Multi-class minimax game based self-supervised tasks
NAS	Neural architecture search
NE	Nash equilibrium
OCR	Optical character recognition
ODE	Ordinary differential equation
ORGAN	Objective-reinforced GAN
RNN	Recurrent neural network
RL	Reinforcement learning
SCH-GAN	Semi-supervised cross-modal hashing GAN
Seq-GAN	Sequence GAN
SGD	Stochastic gradient descent
SNEP	Stochastic nash equilibrium problem
SPE	Subgame perfect equilibrium
SRFB	Stochastic relaxed forward-backward
SS	Self-supervised task
SVI	Stochastic variational inequality
WGAN	Wasserstein GAN

decision maker. In minimax strategy, the decision maker wants to cause harm to others by minimizing other players' maximum payoff (Shoham and Leyton-Brown 2008). The value that players get in the minimax and maximin strategy method is called min–max (minimax) and max–min (maximin) value, respectively. In Neumann (1928), Neumann proved that in any finite, two-player, zero-sum game, all Nash equilibria coincide with min–max and max–min strategies of players. Furthermore, the min–max and max–min values are equal to the Nash equilibrium utility. This is the most crucial result of zero-sum games that applies to generative adversarial networks. Games and more specifically zero-sum games

can be extended to multiplayer mode. There are several ways to determine or approximately determine the NE of multi-player zero-sum games. For example, the authors in Cai and Daskalakis (2011) and Lv and Ren (2018) have presented different models to find the equilibrium solutions of these games.

## 2.2 Generative adversarial networks

In this section, we give a brief description of the GAN by reviewing its basic idea, learning methods, and challenges.

### 2.2.1 Generative models

The main purpose of a generative model is to generate synthetic data whose distribution captures the distribution of a real data set (i.e., the training set). Generative models can be divided into three types. In the first type, the model is trained using a training set with distribution  $p_{data}$  (unknown to model). The model generates data of distribution  $p_{model}$ , which is an approximation of  $p_{data}$ . In the second type, the model is solely capable of producing samples from  $p_{model}$ . The third type can do both. GANs concentrate on the second type, but it is also possible to design third type's GANs (Goodfellow 2016).

### 2.2.2 GAN fundamentals

In 2014, Goodfellow et al. (2014a) introduced GANs as a framework in which two neural network (as players) play a zero-sum game. In this game, the players are called the generator  $G$  and the discriminator  $D$ . The generator is the one that produces the samples while the discriminator tries to distinguish training samples from the generator's samples. The more indistinguishable the samples produced are, the better is the generative model (Goodfellow et al. 2014a). Any differentiable function, such as a multi-layer neural network, can represent generator and discriminator. The generator, denoted by  $G(z)$ , takes as input a prior noise distribution ( $p_z$ ) and maps it to approximate training data distribution ( $p_g$ ). Discriminator,  $D(x)$ , is a mapping of the input data distribution  $p_{data}$  into a real number in the interval of  $[0, 1]$ , which represents the probability that the sample is real and not a fake one (i.e., produced by the generator) (Gonog and Zhou 2019).

### 2.2.3 GAN learning models

The generator and discriminator can be trained using an iterative process for optimizing an objective function. Goodfellow et al. used the following objective function in their game (Goodfellow et al. 2014a):

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)} [f_0(D(x))] + \mathbb{E}_{z \sim p_z(z)} [f_1(1 - D(G(z)))] \quad (1)$$

where  $f_0$  and  $f_1$  take different forms, depending on the specific divergence metrics, as shown in Table 2. The first proposed GAN uses the Jensen-Shannon divergence metric.

To train the simple model, shown in Fig. 1, we first fix  $G$  and optimize  $D$  to make it discriminate as accurately as possible. Next, we fix  $D$  and try to minimize the objective function. At the NE, discriminator cannot distinguish between real and fake data. For example, under the Jensen-Shannon divergence metric, when  $p_r(x)/(p_r(x) + p_r(z)) = 0.5$ , if both

**Table 2** Variant of GANs based on different divergence metrics (Ge et al. 2018)

Divergence metric	$f_0(D)$	$f_1(D)$	Game value
Kullback–Leibler	$\log(D)$	$1 - D$	0
Reverse KL	$-D$	$\log(D)$	$-1$
Jensen–Shannon	$\log(D)$	$\log(1 - D)$	$-\log 4$
WGAN	$D$	$-D$	0

discriminator and generator work optimally, the game reaches the Nash equilibrium and the min-max and max-min values are the same (i.e.,  $-\log 4$  as shown in Table 2).

Finding mixed NE may be impossible since the GANs' game can be represented as an endless game, as we will explain later. The fact that each generator or discriminator seeks to determine the best response to the action of the previous discriminator or generator, or, in other words, find a Subgame Perfect Nash Equilibrium, makes the GANs' game a sequential game from a different point of view. In Farnia and Ozdaglar (2020), the authors showed that GANs might have no Nash Equilibrium. Consequently, various methods are used in order to help us converge to the NE from different perspectives. For multi-player GANs, though they are unable to reach the global optimum, numerous modes have been proposed as a result of various generator initial states (Lv and Ren 2018; Farnia and Ozdaglar 2020).

### 3 Related surveys

As GANs became increasingly popular, numerous surveys have been presented (about 40 surveys so far), which can be classified into three categories. The literatures in the first category (Goodfellow 2016; Wang et al. 2017; Creswell et al. 2018; Hitawala 2018; Gonog and Zhou 2019; Hong et al. 2019; Bissoto et al. 2019; Zhang et al. 2018a; Pan et al. 2019; Kumar and Jayagopal 2021; Salehi et al. 2020; Ghosh et al. 2020; Saxena and Cao 2020; Gui et al. 2020; Jabbar et al. 2020) explore a relatively broad scope in GANs, including key concepts, algorithms, applications, different variants and architectures. In contrast, the surveys in the second group (Lucic et al. 2017; Alqahtani et al. 2021; Wiatrak and Albrecht 2019; Lee and Seok 2020; Pan et al. 2020) focus on a specific issue in GANs (e.g., regularization methods, loss functions, etc) and address how researchers deal with such an issue. In the third category (Cao et al. 2018; Wang et al. 2019, 2020; Wu et al. 2017; Sorin et al. 2020; Tschuchnig et al. 2020; Agnese et al. 2020; Jain and Jayaswal 2020; Sampath et al. 2021; Yi et al. 2019; Yinka-Banjo and Ugot 2020; Ghosh et al. 2020; Di Mattia et al. 2019; Geogres-Filteau and Cirillo 2020; Gao et al. 2020; Shin et al. 2020) the surveys summarize the applications of GAN in a specific field, from computer vision and image synthesis to cybersecurity and anomaly detection. In the following, we briefly review surveys in each category and explain how our paper differs from them.

#### 3.1 GAN general surveys

In Goodfellow (2016), the author provides answers to the most frequently raised questions about GANs. Wang et al. (2017) reviewed theoretic and implementation models of GANs, their applications, and the advantages and disadvantages of this generative model. Creswell

et al. (2018) provided an overview of GANs, especially for the signal processing community, by characterizing different methods for training and constructing GANs, and challenges in the theory and applications. In Ghosh et al. (2020) Ghosh et al. presented a comprehensive summary of the progression and performance of GANs along with their various applications. Saxena and Cao (2020) conducted a survey of the advancements in GANs design, and optimization solutions proposed to handle GANs challenges. Kumar and Jayagopal (2021) presented state-of-the-art research on GANs, their applications, evaluation metrics, challenges, and benchmark datasets. In Salehi et al. (2020) two new deep generative models, including GA, were compared, and the most remarkable GAN architectures were categorized and discussed.

Gui et al. (2020) provided a review of various GANs methods from an algorithmic, theoretic, and applicative perspectives. Jabbar et al. (2020) reviewed different GAN variants, applications, and several training solutions. Hitawala in Hitawala (2018) presented different versions of GANs and provided a comparison between them in some aspects, such as learning, architecture, gradient updates, objective function, and performance metrics. Similarly, Gonog et al. in Gonog and Zhou (2019) reviewed the extensional variants of GANs, and classified them in terms of how they optimized the original GAN or changed its basic structure, as well as their learning methods. In Hong et al. (2019) Hong et al. discussed the details of the GAN from the perspective of various object functions, architectures, and the theoretical and practical issues in training the GANs. The authors also enumerate the GAN variants applied in different domains. Bissoto et al. (2019) conducted a review of GAN advancements on six fronts, including architectural contributions, conditional techniques, normalization and constraint contributions, loss functions, image-to-image translations, and validation metrics. Zhang et al. (2018a) surveyed twelve extended GAN models and classified them in terms of the number of game players. Pan et al. (2019) analyzed the differences among different generative models and classified them from the perspective of architecture and objective function optimization. They also discussed the training tricks and evaluation metrics and presented GANs applications and challenges.

### 3.2 GAN challenges

In the second group of surveys, Lucic et al. (2017) conducted an empirical comparison of GAN models, with a focus on unconditional variants.

Alqahtani et al. (2021) focused on potential applications of GANs in different domains. They attempted to identify advantages, disadvantages and major challenges for successful implementation of GANs. Wiatrak and Albrecht (2019) surveyed current approaches for stabilizing the GAN training procedure, and categorizing various techniques and key concepts. In Lee and Seok (2020), Lee et al. reviewed the regularization methods used in the stable training of GANs, and classified them into several groups by their operation principles. Pan et al. (2020) performed a survey for the loss functions used in GANs, and analyzed the pros and cons of these functions. As differentially private GAN models provide a promising direction for generating private synthetic data, Fan et al. (2020) surveyed the existing approaches presented for this purpose.

### 3.3 GAN applications

The authors in (Cao et al. 2018; Wang et al. 2019, 2020; Wu et al. 2017; Sorin et al. 2020; Tschuchnig et al. 2020; Agnese et al. 2020; Jain and Jayaswal 2020; Sampath



et al. (2021; Yi et al. 2019) conducted reviews of different aspects of GAN progress in the field of computer vision and image synthesis.

Cao et al. (2018) reviewed recent GAN models and their applications in computer vision. Cao et al. (2018) compared the classical and state-of-the-art GAN algorithms in terms of the mechanism and visual results of generated samples. Wang et al. (2019) structured a review toward addressing practical challenges relevant to computer vision. They discussed the most popular architecture-variant, and loss-variant GANs, for tackling these challenges. Wu et al. (2017) presented a survey of image synthesis, image editing, and video generation with GANs. They covered recent papers that leverage GANs in image applications, including texture synthesis, image inpainting, image-to-image translation, image editing, and video generation. Along the same lines, Wang et al. (2020) reviewed the recent research on GANs in the field of image processing, and categorized them in four fields including image synthesis, image-to-image translation, image editing, and cartoon generation.

Surveys such as Agnese et al. (2020) and Jain and Jayaswal (2020) focused on reviewing recent techniques to incorporate GANs in the problem of text-to-image synthesis. In Agnese et al. (2020), Agnese et al. proposed a taxonomy to summarize GAN-based text-to-image synthesis papers into four major categories: Semantic Enhancement GANs, Resolution Enhancement GANs, Diversity Enhancement GANs, and Motion Enhancement GANs. Different from the other surveys in this field, Sampath et al. (2021) examined the most recent developments of GANs techniques for addressing imbalance problems in image data. The real-world challenges and implementations of synthetic image generation based on GANs are covered in this survey.

In Yi et al. (2019); Sorin et al. (2020); Tschuchnig et al. (2020), the authors dealt with the medical applications of image synthesis by GANs. Yi et al. (2019) described the promising applications of GANs in medical imaging and identified some remaining challenges that need to be solved. Another paper on this subject, Sorin et al. (2020) reviewed GANs' application in image denoising and reconstruction in radiology. Tschuchnig et al. (2020) summarized existing GAN architectures in the field of histological image analysis.

The authors of Yinka-Banjo and Ugot (2020) and Ghosh et al. (2020) provided reviews on the application of GANs in cybersecurity. Yinka-Banjo and Ugot (2020) surveyed studies where the GAN plays a key role in the design of security or adversarial system. Ghosh et al. (2020) focused on the various ways in which GANs have been used to provide both security advances and attack scenarios to bypass detection systems.

Di Mattia et al. (2019) surveyed the principal GAN-based anomaly detection methods. Georges-Filteau and Cirillo (2020) reviewed the published literature on Observational Health Data to uncover the reasons for the slow adoption of GANs for this subject. Gao et al. (2020) addressed the practical applications and challenges relevant to spatio-temporal applications, such as trajectory prediction, events generation, and time-series data imputation. The recently proposed user mobility synthesis schemes based on GANs are summarized in Shin et al. (2020).

According to the classification provided for review papers, our survey falls into the second category. We focus specifically on the recent progress of the application of game-theoretical approaches toward addressing the GAN challenges. While several surveys for GANs have been presented to date, to the best of our knowledge, our survey is the first to address this topic. Although the authors in Wiatrak and Albrecht (2019) presented a few game-model GANs, they have not done a comprehensive survey in this field, and many new pieces of research have not been covered. We hope that our survey will serve as a

reference for interested researchers on this subject, who would like to extend GANs with leverage on game-theoretical approaches.

## 4 Games of GANs: a taxonomy

In this section, we present a taxonomy for classifying GANs into three main categories, by focusing on how these works extended the original GAN As we mentioned in Sect. 2.1, three main elements that make a game are players, strategy sets, and payoffs. Changing each of these elements can affect the game results such as equilibrium points and the players' profits. As GANs basically model a game between the generator and the discriminator, these changes cause the result and metrics of GANs to be affected. Different papers have utilized these changes to mitigate the mode collapse, or improve GAN's stability and convergence. In this regard, in our taxonomy, we have considered which game elements have changed and how literature utilized these factors to improve GAN features.

Our taxonomy is done in terms of (1) *modified game mode*, (2) *architecture modification*, and (3) *modified learning algorithms*. In the first category, we focus on the literature that models the interaction between the generator and the discriminator by the other game types, such as Stackelberg rather than a basic model which is zero-sum. Modifying the game model, which is considered in this category, changes the strategy chosen by the players at the Nash equilibrium points, and in subsequent, the equilibrium points of GANs. In the second category, we review papers that have changed the GAN architecture in a way that extends the idea of using a single pair of the generator and the discriminator to the multi-agent setting. This change leads the two-player game to transform into the multiple games or multi-agent games. The GAN's structure is important as it has a significant impact on the stability and also the performance. In the third category, we cover articles that changed the learning algorithm of the players from a best-response in the original GAN model to other ones, such as No-regret, or Fictitious play. As we mentioned in Sect. 2, generally, in game theory, a player chooses a strategy which is the best response given the other players' strategies. However players can change their strategies by applying other learning algorithms. For example, instead of maximizing their profit in the best-response case, choose strategies that minimize their regret by a regret minimization algorithm.

We further classify each category into several subsets, as shown in Fig. 3. In the following subsections, we explain each category in detail and discuss the recent advances in each group.

The authors in the reviewed papers have used different metrics to evaluate their proposed GAN models. Table 3 presents all these metrics and indicates which ones were used in each of the papers. Metrics range from quantitative to qualitative ones. The most-reported metrics are Inception Score (IS) and Fréchet Inception Distance (FID). However, metrics are not limited to these ones. Given recent progress in GAN and its application, new measures have emerged. Borji reviews about 30 quantitative and qualitative measures for evaluating generative models with an emphasis on GAN-derived models, and discusses their pros and cons in Borji (2019).



**Fig. 3** Proposed taxonomy of GAN advances by game theory

**Table 3** Metrics which are reported for results in the reviewed papers

Metrics	Reference
IS	Stackelberg GAN Zhang et al. (2018c), Farnia and Ozdaglar (2020), Franci and Grammatico (2020), MGAN Hoang et al. (2018), DDL-GAN Jin et al. (2020), GMAN Durugkar et al. (2016), MD-GAN Hardy et al. (2019), Arora et al. (2017), DRAGAN Kodali et al. (2017), Fictitious GAN Ge et al. (2018), Tian et al. (2020), D2GAN Nguyen et al. (2017)
FID	Stackelberg GAN Zhang et al. (2018c), DDL-GAN Jin et al. (2020), Microbatch GAN Mordido et al. (2020), MD-GAN Hardy et al. (2019), Tran et al. (2019), FedGAN Rasouli et al. (2020), Tian et al. (2020)
Chi-square	MAD-GAN Ghosh et al. (2018)
KL	MAD-GAN Ghosh et al. (2018)
Reverse KL	Chekhov GAN Grnarova et al. (2017)
P-value	SeqGAN Yu et al. (2017)
Human scores	Diversity-promoting GAN Xu et al. (2018), SeqGAN Yu et al. (2017)
Classification scores	Fan and Liu (2020), Zhang et al. (2018a), Aghakhani et al. (2018), CS-GAN Li et al. (2018)
Others	Fan and Liu (2020), ORGAN Guimaraes et al. (2017), OptiGAN Hossam et al. (2020)

## 4.1 Modified game model

As we mentioned in Sect. 2.2.2, the core of all GANs is a competition between a generator and a discriminator, which are modeled as a game. Therefore, game theory plays a key role in this context. However, most GANs relying on the basic model, formulating it as a zero-sum (minimax) game, but some research utilized other game variants to tackle the GANs' challenges. In this section, we review these literature. Based on the proposed methods, we classify the works under this category into three subcategories. Section 4.1.1 presents researches that cast the training process as a stochastic game. Research works presented in Sect. 4.1.2 apply the idea of leader-follower of the Stackelberg game in the GANs. Finally Sect. 4.1.3 presents GANs models as a Bi-affine game. A summary of the reviewed researches in the modified game model category is shown in Table 4.

### 4.1.1 Stochastic game

One of the main issues for GANs is that these neural networks are very hard to train because of the convergence problems. Franci et al. in Franci and Grammatico (2020) addressed this problem by casting the training procedure as a *stochastic Nash equilibrium problem* (SNEP). The SNEP will recast as a *stochastic variational inequality* (SVI) and target the solutions that are SNE. The advantage of this approach is that there are many algorithms for finding the solution of an SVI, like the forward-backward algorithm, also known as gradient descent. Franci et al. proposed a *stochastic relaxed forward-backward* (SRFB) algorithm and a variant with an additional step for averaging over decision variables (aSRFB) for the training process of GANs. For proving convergence to a solution, we need monotonicity on the pseudogradient mapping, which is defined by Eq. (2), where  $J_g$  and  $J_d$  are the payoff functions of the generator and the discriminator.

**Table 4** Summary of publications included in the modified game model category (Sect. 4.1)

Reference	Convergence	Methodology and contributions	Pros	Cons
Stackelberg Game: Sect. 4.1.2				
Stackelberg GAN Zhang et al. (2018c)	SPNE	Models multi-generator GANs as a Stackelberg game	Can be built on top of standard GANs and proved the convergence	–
Farnia and Ozdaglar (2020)	SPE	Theoretical examples of standard GANs with no NE and proximal equilibrium as a solution, proximal training for GANs	Can apply to any two-player GANs, allow the discriminator to locally optimize	Focus only on the pure strategies of zero-sum GANs in non-realizable settings
Stochastic Game: Sect. 4.1.1				
Franci and Grammatico (2020)	SNE	Cast the problem as SNEP and recast it to SVI, SRFB and aSRFB solutions	Proved the convergence	Need monotonicity on the pseudogradient mapping, increasing number of samples to reach an equilibrium
Bi-affine Game: Sect. 4.1.3				
Hsieh et al. (2019)	Mixed NE	Tackling the training of GANs by reconsidering the problem formulation from the mixed Nash Equilibria perspective	Showing that all GANs can be relaxed to mixed strategy forms, flexibility	–

$$\mathbb{F} = \begin{bmatrix} \mathbb{E}[\nabla_{x_g} J_g(x_g, x_d)] \\ \mathbb{E}[\nabla_{x_d} J_d(x_d, x_g)] \end{bmatrix} \quad (2)$$

If pseudogradient mapping of the game is monotone and the increasing number of samples is available, the algorithm converges to the exact solution but with only finite, fixed mini-batch samples, and by using the averaging technique, it will converge to a neighborhood of the solution.

#### 4.1.2 Stackelberg game

One of the main issues for GAN is the convergence of the algorithm. Farnia et al. in Farnia and Ozdaglar (2020) showed that "GAN zero-sum games may not have any local Nash equilibria" by presenting certain theoretical and numerical examples of standard GAN problems. Therefore, based on the natural sequential type of GANs, where the generator moves first and follows the discriminator (leader), this problem can be considered as a Stackelberg game and focused on *subgame perfect equilibrium* (SPE). For solving the convergence issue, the authors tried to find the equilibrium called *proximal equilibrium* which enables traversing the spectrum between Stackelberg and Nash equilibria. In a proximal equilibrium, as shown in Eq. (3), we allow the discriminator to optimize locally in a norm-ball nearby the primary discriminator. To keep the  $\tilde{D}$  close to  $D$ , they penalize the distance among the two functions by  $\lambda$ , as  $\lambda$  goes from zero to infinity, the equilibria change from Stackelberg to Nash.

$$V_{\lambda}^{prox}(G, D) := \max_{\tilde{D} \in D} V(G, \tilde{D}) - \frac{\lambda}{2} \|\tilde{D} - D\|^2 \quad (3)$$

Farnia et al. also proposed proximal training which optimizes the proximal objective  $V_{\lambda}^{prox}(G, D)$  instead of the original objective  $V(G, D)$  that can apply to any two-player GAN. With respect to dimensions of the network, the proximal training improves WGAN-WC from 10 to 60%. Zhang et al. (2018c) also used Stackelberg game and presented Stackelberg GAN to tackle the instability of GANs training process. Stackelberg GAN is using a multi-generator architecture and the competition is between the generators (followers) and the discriminator (leader). We discussed the architecture details in Sect. 4.2.1.

#### 4.1.3 Bi-affine game

Hsieh et al. in Hsieh et al. (2019) examined training of GANs by reconsidering the problem formulation from the mixed NE perspective. In the absence of convexity, the theory focuses only on the local convergence, and it implies that even the local theory can break down if intuitions are blindly applied from convex optimization. In Hsieh et al. (2019) the mixed Nash Equilibria of GANs are proposed. They are global optima of infinite-dimensional bi-affine games. Finite-dimensional bi-affine games are also applied for finding mixed NE of GANs. They also show that we can relax all current GAN objectives into their mixed strategy forms. Eventually, in this article, it's experimentally shown that their method achieves better or comparable performance than popular baselines such as SGD, Adam, and RMSProp.

## 4.2 Modified architecture

As we mentioned in Sect. 2, GAN is a framework for producing a generative model through a two-player minimax game; however, in recent works, this idea is extended by using more than two players.

In this section, we review literature in which proposed GAN variants have modified the architecture in such a way that we have GANs with a mixture of generators, and/or discriminators and show how applying such methods can provide better convergence properties and prevent mode collapse. The majority of the works in this category focus on introducing a larger number of generators and/or discriminators. But, in some papers, the number of generators and discriminators has not been changed, and a new agent has been added converting the problem into a multi-agent scenario.

In Sect. 4.2.1, we will discuss GAN variants which extended the basic structure from a single generator to many generators. In Sect. 4.2.2, we are going to review articles that deal with the problem of mode collapse by increasing the number of discriminators in order to force the generator to produce different modes. Section 4.2.3 is dedicated to discussing papers that develop GANs with multiple generators and multiple discriminators. Articles will be reviewed in Sects. 4.2.4 and 4.2.5 extend the architecture by adding another agent, which is a classifier (Sect. 4.2.4) or an RL agent (Sect. 4.2.5), to show the benefits of adding these agents to GANs. The methodologies, contributions as well as the pros and cons of reviewed papers are summarized in Table 5.

### 4.2.1 Multiple generators, one discriminator

The minimax gap is smaller in GANs with multi-generator architecture and more stable training performances are experienced in these GANs (Zhang et al. 2018c). As we mentioned in Sect. 4.1.2, Zhang et al. (2018c) tackled the problem of instability during the GAN training as a result of a gap between minimax and maximin objective values. To mitigate this issue, they designed a multi-generator architecture and modeled the competition among agents as a Stackelberg game. Results have shown the minimax duality gap decreases as the number of generators increases. In this article, the mode collapse issue is also investigated and showed that this architecture effectively alleviates the mode collapse issue. This model reduces the FID by 21.52% compared to Ghosh et al. (2018) and 14.61% compared to Hoang et al. (2018). One of the significant advantages of this architecture is that it can be applied to all variants of GANs, e.g., Wasserstein GAN, vanilla GAN, etc. Additionally, with an extra condition on the expressive power of generators, it is shown that Stackelberg GAN can achieve  $\epsilon$ -approximate equilibrium with  $\tilde{O}(1/\epsilon)$  generator Zhang et al. (2018c).

Furthermore, Ghosh et al. (2018) proposed a multi-generator and single discriminator architecture for GANs named Multi-Agent Diverse Generative Adversarial Networks (MAD-GAN). In this paper, different generators capture varied, high probability modes, and the discriminator is designed such that, along with finding the real and fake samples, identifies the generator that generated the given fake sample (Ghosh et al. 2018). It is shown that at convergence, the global optimum value of  $-(k+1)\log(k+1) + k\log k$  is achieved, where  $k$  is the number of generators.

Comparing the presented models in Ghosh et al. (2018) and Zhang et al. (2018c), in MAD-GAN (Ghosh et al. 2018) multiple generators are combined with the assumption that the generators and the discriminator have infinite capacity, but in the Stackelberg

GAN (Zhang et al. (2018c) there is no assumption on the model capacity. Also, in MAD-GAN (Ghosh et al. 2018) the generators share common network parameters, although, in the Stackelberg GAN (Zhang et al. 2018c) various sampling schemes beyond the mixture model are allowed, and each generator has free parameters.

The assumption that increasing generators will cover the whole data space is not valid in practice. So Hoang et al. (2018), in contrast with Ghosh et al. (2018), approximated data distribution by forcing generators to capture a subset of data modes independently of those of others instead of forcing generators by separating their samples. Thus, they established a minimax formulation among a classifier, a discriminator, and a set of generators. The classifier determines which generator generates the sample by performing multi-class classification. Each generator is encouraged to generate data separable from those produced by other generators, because of the interaction between generators and the classifier. In this model, multiple generators create the samples. Then one of them will be randomly picked as the final output similar to the mechanism of a probabilistic mixture model. Therefore they theoretically proved that, at the equilibrium, the Jensen–Shannon divergence (JSD) between the final output and the data distribution is minimal. In contrast, the JSD amongst generators' distributions is maximal, hence the mode collapse problem is effectively avoided. Moreover, the computational cost that is added to the standard GAN is minimal in the suggested model by applying parameter sharing. The proposed model can efficiently scale to large-scale datasets as well and improve the Inception Score by about 16% to 38% compared to Nguyen et al. (2017) and Durugkar et al. (2016), respectively.

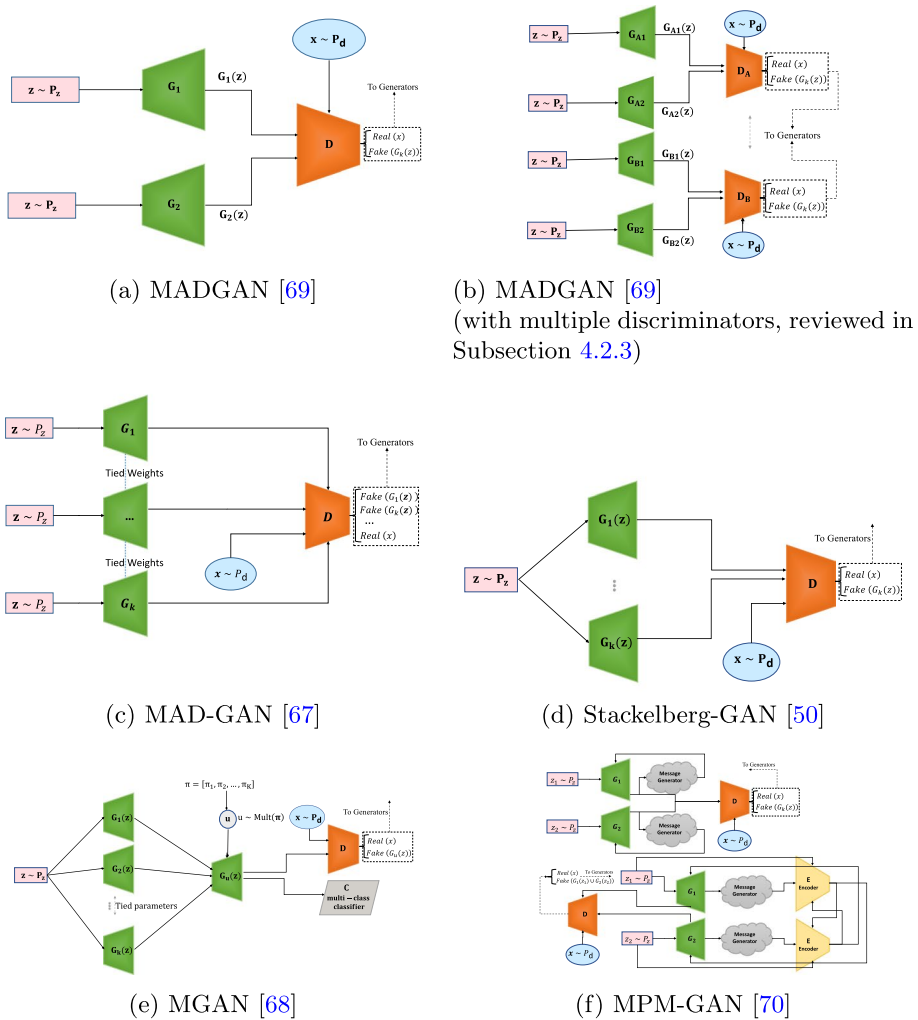
Ke and Liu (2020) proposed a new architecture, named multiagent distributed GAN (MADGAN). In this framework, the discriminator is considered a leader, and the generator is considered a follower. Moreover, this proposal is based on the social group wisdom and the influence of the network structure on agents. MADGAN can have a multi-generator and multi-discriminator architecture (e.g., two discriminators and four generators) as well as multiple generators and single discriminator architecture, which is our discussed topic in this section. As one of the vital contributions of MADGAN, it can train multiple generators simultaneously, and the training results of all generators are consistent.

Furthermore, in Message Passing Multi-Agent Generative Adversarial Networks (Ghosh et al. 2016b), the authors proposed that with two generators and one discriminator, communicating through message passing, better image generation can be achieved, and the amount of classification error can be reduced by about 24% compared to DCGAN. In this paper, there are two objectives such as competing and conceding. The competing is based on the fact that the generators compete with each other to get better scores for their generated data from the discriminator. However, the conceding is based on the fact that the two generators try to guide each other in order to get better scores for their generations from the discriminator and ensure that the message sharing mechanism guides the other generator to generate better than itself. Generally, in this section, innovative architectures and objectives aimed at training multi-agent GANs are presented. Figure 4 presents a schematic view of GANs' structure reviewed in this section.

#### 4.2.2 One generator, multiple discriminators

The multi-discriminators are constructed with homogeneous network architecture and trained for the same task from the same training data. In addition to introducing a multi-discriminators schema, Durugkar et al. (2016), from the perspective of game theory, showed that discriminators act like each other; thus, they will converge to similar decision





**Fig. 4** A schematic view of GAN variants with multiple generators and one discriminator, presented in Sect. 4.2.1. D: Discriminator, G: Generator,  $z$ : Noise,  $x$ : Real data,  $P_z$ : Latent space,  $P_x$ : Data distribution,  $G_i(z)$ : Fake data generated by generator  $i$ , C: Classifier, E: Encoder

boundaries. In the worst case, they may even converge to a single discriminator. So, Jin et al. (2020) by discriminator discrepancy loss (DDL), the multiplayer minimax game unifies the optimization of DDL and the GAN loss, seeking an optimal trade-off between the accuracy and diversity of multi discriminators. In addition, improve the IS and FID by 1% to 3% in comparison to Durugkar et al. (2016). Compared to Durugkar et al. (2016), Hardy et al. (2019) distributed discriminators over multiple servers. Thus, they can train over datasets that are spread over numerous servers.

Aghakhani et al. (2018) proposed FakeGAN. The GAN model uses two discriminators and one generator. The discriminators use the Monte Carlo search algorithm to evaluate and pass the intermediate action-value as the reinforcement learning (RL) reward to the generator. The generator is modeled as a stochastic policy agent in RL

(Aghakhani et al. 2018). Instead of one batch in Jin et al. (2020), Mordido et al. (2020) divided generated samples into multiple micro-batch. Then update each discriminator's task to discriminate between different samples. Samples coming from its assigned fake micro-batch and samples from the micro-batches assign to the other discriminator, together with the real samples. This proposed idea improved the IS by 10% to 15% compared to Durugkar et al. (2016).

Unlike Durugkar et al. (2016), Nguyen et al. (2017) combined the Kullback–Leibler (KL) and reverse KL divergence (the measure of how one probability distribution is different from a second) into a unified objective function. Combining these two measures can exploit the divergence's complementary statistical properties to diversify the estimated density in capturing multi modes effectively cover almost 59% more modes than Goodfellow et al. (2014b). From the perspective of game theory in Nguyen et al. (2017), there are two discriminators and one generator with the analogy of a three-player minimax game. In this case, there exist two pairs of players which are playing two minimax games simultaneously. In one of the games, the discriminator rewards high scores for samples from data distribution (i.e., reverse KL divergence, in Eq. (4)), while another conversely rewards high scores for samples from the generator, and the generator produces data to fool two discriminators (i.e., KL divergence, in Eq. (5)).

$$\min_G \max_{D_1} J(G, D_1) = \alpha \times \sum_{x \sim P_{data}} [\log D_1(x)] + \sum_{z \sim P_z} [-D_1(G(z))], \quad (4)$$

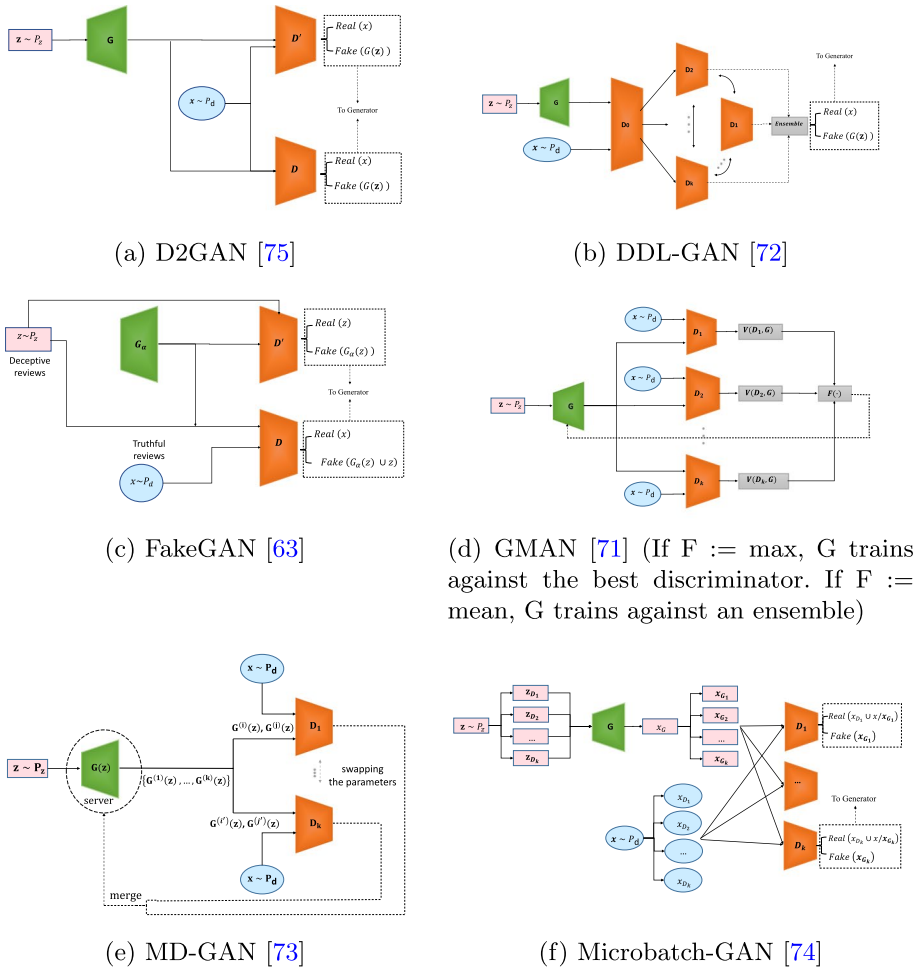
$$\min_G \max_{D_2} J(G, D_2) = \sum_{x \sim P_{data}} [-D_2(x)] + \beta \times \sum_{z \sim P_z} [\log D_2(G(z))], \quad (5)$$

where in the above equations hyperparameters  $\alpha, \beta$  are being used to control and stabilize the learning method.

Minimizing the Kullback–Leibler (KL) divergence between data and model distributions covers multiple modes but may produce completely unseen and potentially undesirable samples. In reverse KL divergence, it is observed that optimization towards the reverse KL divergence criteria mimics the mod seeking process where the  $P_{model}$  concentrates on a single mode of  $P_{data}$  while ignoring other modes. Figure 5 presents a schematic view of the GANs reviewed in this section.

### 4.2.3 Multiple generators, multiple discriminators

The existence of equilibrium has always been considered one of the open theoretical problems in this game between generator and discriminator. Arora et al. (2017) turned to infinite mixtures of generator's deep nets in order to investigate the existence of equilibria. Unsurprisingly, equilibrium exists in an infinite mixture. Therefore, they showed that a mixture of a finite number of generators and discriminators can approximate the min-max solution in GANs. This implies that an approximate equilibrium can be achieved with a mixture (not too many) of generators and discriminators. They also proposed a heuristic approximation to the mixture idea to introduce a new framework for training called MIX+GAN: use a mixture of T components, where T is as large as allowed by the size of GPU memory (usually  $T \leq 5$ ). In fact, a mixture of T generators and T discriminators are trained which share the same network architecture but have their own trainable parameters. Maintaining a mixture represents maintaining a weight  $w_{u_i}$  for the generator  $G_{u_i}$  which corresponds to the probability of selecting the output of  $G_{u_i}$ . These weights for the generator are updated



**Fig. 5** Schematic view of GANs with one generator and multiple discriminators, reviewed in Sect. 4.2.2.  $D$ : Discriminator,  $G$ : Generator,  $z$ : Noise,  $x$ : Real data,  $P_z$ : Latent space,  $P_x$ : Data distribution,  $G(z)$ : Fake data generated by the generator

by backpropagation. This heuristic can be applied to the existing methods like DCGAN, W-GAN, etc. Experiments show that the MIX + GAN protocol improves the quality of several existing GAN training methods and can lead to more stable training.

As we mentioned earlier, one of the significant challenges in GAN algorithms is their convergence. Referring to Rasouli et al. (2020), this challenge is a result of the fact that cost functions may not converge using gradient descent in the minimax game between the discriminator and the generator. Convergence is also one of the considerable challenges in federated learning. This problem becomes even more challenging when data at different sources are not independent and identically distributed. Therefore, Rasouli et al. proposed an algorithm for multi-generator and multi-discriminator architecture for training a GAN with distributed sources of non-independent-and-identically-distributed data sources named Federated Generative Adversarial Network (FedGAN) (Rasouli

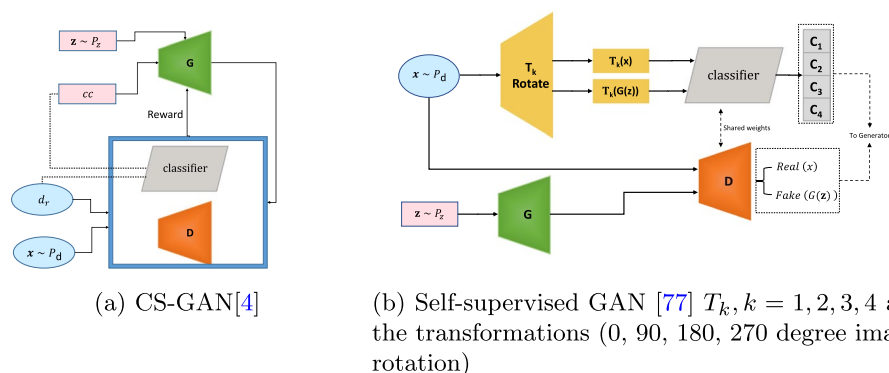
et al. 2020). Local generators and discriminators are used in this algorithm. These generators and discriminators are periodically synchronized via an intermediary that averages and broadcasts the generator and discriminator parameters. In fact, results from stochastic approximation for GAN convergence and communication-efficient SGD for federated learning are connected in this article to address FedGAN converge. One of the notable results in Rasouli et al. (2020) is that FedGAN has similar performance to general distributed GAN while it converges and reduces communication complexity as well.

In Ke and Liu (2020) proposed a multi-agent distributed GAN (MADGAN) framework based on the social group wisdom and the influence of the network structure on agents, in which the discriminator and the generator are regarded as the leader and the follower, respectively. The multi-agent cognitive consistency problem in the large-scale distributed network is addressed in MADGAN. In Ke and Liu (2020) the conditions of consensus are presented for a multi-generator and multi-discriminator distributed GAN by analyzing the existence of stationary distribution to the Markov chain of multiple agent states. The experimental results show that the generation effect of the generators trained by MADGAN can be comparable to that of the generator trained by GAN. Moreover, MADGAN can train multiple generators simultaneously, and the training results of all generators are consistent.

#### 4.2.4 One generator, one discriminator, one classifier

One of the issues that GANs face is catastrophic forgetting in the discriminator neural network. Self-Supervised (SS) tasks were planned to handle this issue, however, these methods enable a seriously mode-collapsed generator to surpass the SS tasks. In Tran et al. (2019), Tran et al. proposed new SS tasks, called Multi-class minimax game based Self-supervised tasks (MS) which are based on a multi-class minimax game, including a discriminator, a generator, and a classifier. The SS task is a 4-way classification task of recognizing one among the four image rotations (i.e., 0, 90, 180, and 270 degrees). The discriminator SS task is to train the classifier C that predicts the rotation applied to the real samples and the generator SS task is to train the generator G to produce fake samples for maximizing classification performance. The SS task helps the generator learn the data distribution and generate diverse samples by closing the gap between supervised and unsupervised image classification. The theoretical and experimental analysis showed that the convergence of this approach has progressed.

The authors in Mullick et al. (2019) and Dam et al. (2022) focused on the class imbalance classification problem and get helped from GAN to propose a method for handling this challenge. Mullick et al. seek to build a classifier that can work with an imbalanced dataset in Mullick et al. (2019). To make such a classifier, they use a convex generator that generates new samples from the minority classes as convex combinations of existing instances. They model the interaction between the convex generator, the multi-class classifier that aims to correctly classify the data points, and a conditional discriminator as a three-player adversarial game. Different from the existing GAN models where the generator works in harmony with the classifier to fool the discriminator, in this model, the generators attempts to fool both the discriminator and the classifier into misclassifying the generated samples. Dam et al. present an end-to-end deep generative classifier in Dam et al. (2022). They propose a domain-constraint AE to preserve the latent-space as prior for a generator, which plays an adversarial game with two other deep networks, a discriminator



**Fig. 6** Schematic view of GANs with one generator, one discriminators, and one classifier reviewed in Sect. 4.2.4. D: Discriminator, G: Generator,  $z$ : Noise,  $x$ : Real data,  $P_z$ : Latent space,  $P_x$ : Data distribution,  $G(z)$ : Fake data generated by the generator

and a classifier. The authors replaced convex generators used in Mullick et al. (2019) with a neural network generator.

In Li et al. (2018), Liu et al. also used a classifier to generate categorized text. The authors proposed a new framework Cyclic-Synthesized GAN (CS-GAN), which uses GAN, RNN, and RL to generate better sentences. The classifier position is to ensure that the generated text contains the label information and the RNN is a character predictor because the model is built at the character level to limit the large searching space. We can divide the generation process into two steps, first adding category information into the model and making the model generate category sentences respectively, then combining category information in GAN to generate labeled sentences. CS-GAN acts strongly in supervised learning, especially in the multi-categories datasets. Figure 6 presents the architecture of GANs reviewed in this subsection.

#### 4.2.5 One generator, one discriminator, one RL agent

With an RL agent, we can deploy fast and robust control over the GAN's output or input. Moreover, such architecture can be used to optimize the generation process by adding an arbitrary (not necessarily differentiable) objective function to the model.

In De Cao and Kipf (2018), Cao et al. used this architecture for generating molecules and drug discovery. The authors encoded the molecules as the original graph-based representation, which has no overhead compared to similar approaches like SMILES (Weininger 1998), which generates a text sequence from the original graph. For the training part, the authors were not only interested in generating chemically valid compounds, but also tried to optimize the generation process toward some non-differentiable metrics (e.g., how likely the new molecule is water-soluble or fat-soluble) using an RL agent. In Molecular GAN (MolGAN), an external software will compute the RL loss for each molecule. The linear combination of RL loss and WGAN loss is utilized by the generator.

Weininger (1998) tackled the same problem. Compared to De Cao and Kipf (2018), they encoded the molecules as text sequences by using SMILES, the string representation of the molecule, not the original graph-based one. They presented Objective-Reinforced Generative Adversarial Networks (ORGAN), which is built on SeqGAN (Yu et al. 2017).

**Fig. 7** Schematic view of GANs with one generator, one discriminator, and one RL agent reviewed in Sect. 4.2.5. In **b** **G** is trained to maximize a weighted average of the objective, and the discriminator. In **c** the reward network approximate the reward function of a sample and optimize molecule generation using RL. It assigns a reward to each molecule to match a score provided by an external software. *D*: Discriminator; *G*: Generator; *z*: Noise; *x*: Real data;  $P_z$ : Latent space;  $P_X$ : Data distribution;  $G(z)$ : Fake data generated by the generator; *Objectives*: domain-specific metrics

Their RL agent uses REINFORCE (Williams 1992), a gradient-based approach. However, MolGAN (De Cao and Kipf 2018) uses deep deterministic policy gradient (DDPG) (Lillicrap et al. 2015), an off-policy actor-critic algorithm. MolGAN gains better chemical property scores compared to ORGAN, but this model suffers from mode collapse because both the GAN and the RL objective do not encourage generating diverse outputs; alternatively, the ORGAN RL agent depends on REINFORCE, and the unique score is optimized penalizing non-unique outputs.

For controlling the generator, we can also use an RL agent. In Sarmad et al. (2019), Sarmad et al. presented RL-GAN-Net, a real-time completion framework for point cloud shapes. Their suggested architecture is the combination of an auto-encoder (AE), a reinforcement learning (RL) agent, and a latent-space generative adversarial network (l-GAN). Based on the pre-trained AE, the RL agent selects the proper seed for the generator. This idea of controlling the GAN’s output can open up new potentialities to overcome the fundamental instabilities of current deep architectures. Figure 7 presents the structure of GAN variants, described in this section.

**Table 5** Summary of the publication included in the modified architecture category (Sect. 4.2)

Reference	Methodology and contributions	Pros	Cons
Multiple generators, one discriminator: Sect. 4.2.1			
Stackelberg GAN, Zhang et al. (2018c)	Tackling the instability problem in the training procedure with multi-generator architecture	More stable training performances, alleviate the mode collapse	–
MADGAN, Ke and Liu (2020)	A multiagent distributed GAN framework based on the social group wisdom	Simultaneously training of multiple generators, consistency of all generators' training results	–
MAD-GAN, Ghosh et al. (2018)	A multi-agent diverse GAN architecture	Capturing diverse modes while producing high-quality samples	Assumption of infinite capacity for players, global optimal is not practically reachable
MGAN, Hoang et al. (2018)	Encouraging generators to generate separable data by classifier	Overcoming the mode collapsing, diversity	–
MPMGAN, Ghosh et al. (2016b)	An innovative message passing model, where messages being passed among generators	Improvement in image generation, valuable representations from message generator	–
One generator, multiple discriminators: Sect. 4.2.2			
DDL-GAN, Jin et al. (2020)	Using DDL	Diversity	Only applicable to multiple discriminators
D2GAN, Nguyen et al. (2017)	Combine KL, reverse KL	Quality and diversity, scalable for large-scale datasets	Not powerful as the combination of autoencoder or GAN
GMAN, Durugkar et al. (2016)	Multiple discriminators	Robust to mode collapse	Complexity, converge to same outputs
Microbatch GAN, Mordido et al. (2020)	Using microbatch	Mitigate mode collapse	–
MD-GAN, Hardy et al. (2019)	Parallel computation, distributed data	Less communication cost, computation complexity	–
FakeGAN, Aghakhani et al. (2018)	Text classification	–	–
Multiple generators, multiple discriminators: Sect. 4.2.3			
Arora et al. (2017)	Tackling generalization and equilibrium in GANs	Improve the quality of several existing GAN training methods	Aggregation of losses with an extra regularization term, discourages the weights being too far away from uniform

**Table 5** (continued)

Reference	Methodology and contributions	Pros	Cons
MADGAN, Ke and Liu (2020)	Address the multiagent cognitive consistency problem in large-scale distributed network	Simultaneously training of multiple generators, consistency of all generators' training results	–
FedGAN, Rasouli et al. (2020)	A multi-generator and multi-discriminator architecture for training a GAN with distributed sources	Similar performance to general distributed GAN with reduction in communication complexity	–
One generators, one discriminator, one classifier: Sect. 4.2.4 CS-GAN, Li et al. (2018)	Combine RNN, GAN, and RL. Use a classifier to validate category, A character level model	generate sentences based on category, limiting action space	–
Tran et al. (2019)	Multi-class minimax game based self-supervised tasks, use RL agent: Sect. 4.2.5	Improve convergence, can integrate into GAN models	–
One generators, one discriminator, one RL agent: Sect. 4.2.5 MolGAN, De Cao and Kipf (2018)	Use Original graph-structured data, use RL objective to generate specific chemical property	Better chemical property scores, no overhead in representation	Susceptible to mode collapse
ORGAN, Guimaraes et al. (2017)	Encode molecules as text sequences, control properties of generated samples with RL, use Wasserstein distance as loss function	Better result than trained RNNs via MLE or SeqGAN	Overhead in representation, works only on sequential data
RL-GAN-Net, Sarma et al. (2019)	Use RL to find correct input for GAN, Combine AE, RL and l-GAN	A real time point cloud shape completion, less complexity	–



### 4.3 Modified learning algorithm

This category covers methods in which the proposed improvements involve modification in learning methods of GANs from game theory perspective. In more details, in this section, we turn our attention to the literature which combine the other learning approaches of game theory such as fictitious play and reinforcement learning with GANs. It is worth noting that despite the significant progress in the field of learning algorithms, especially in deep learning and the presentation of various learning methods such as continual learning (CL) which have also been applied in or improve GANs (for e.g. Ebrahimi et al. 2020; Mariani et al. 2018; Zhao et al. 2022), our focus in this category is on those learning methods that are used in multi-agent systems which model the interaction between agents as a game.

Different variations of GANs which are surveyed in Sect. 4.3.1 study GAN training process as a regret minimization problem instead of the popular view which seeks to minimize the divergence between real and generated distributions. As another learning method, Sect. 4.3.2 utilizes fictitious play to simulate the training algorithm on GAN. Section 4.3.3 provides a review on the proposed GAN models that are used a federated learning framework which trains across distributed sources to overcome the data limitation of GANs. Researches in Sect. 4.3.4 seek to make a connection between GAN and RL. Table 6 summarizes the contributions, pros and limitations of the literature reviewed in this section.

#### 4.3.1 No-regret learning

The best response algorithms for GAN are often computationally intractable, and they do not lead to convergence and have cycling behavior even in simple games. However, the simple solution, in that case, is to average the iterates. Regret minimization is the more suitable way to think about GAN training dynamics. In Kodali et al. (2017), Kodali et al. proposed studying GAN training dynamics as a repeated game that both players use no-regret algorithms. Moreover, the authors showed that the GAN game's convex-concave case has a unique solution. If  $G$  and  $D$  have enough capacity in the non-parametric limit and updates are made in the function space, the GAN game is convex-concave. The convergence (of averaged iterations) can also be guaranteed using no-regret algorithms. With standard arguments from game theory literature, the authors show that the discriminator does not need to be optimal at each step.

In contrast to Kodali et al. (2017), much of the recent developments (Goodfellow 2016) are based on the unrealistic assumption that the discriminator is playing optimally; this corresponds to at least one player using the best-response algorithm. But in the practical case with neural networks, these convergence results do not hold because the game objective function is non-convex. In non-convex games, global regret minimization and equilibrium computation are computationally hard. Moreover, Kodali et al. (2017) also analyzed GAN training's convergence from this point of view to understand mode collapse. They showed that mode collapse happens because of undesirable local equilibria in this non-convex game (accompanied by sharp gradients of the discriminator function around some real data points). Furthermore, the authors showed that a gradient penalty scheme can avoid the mode collapse by regularizing the discriminator to constrain its gradients in the ambient data space.

**Table 6** Summary of the publication included in the learning method category (Sect. 4.2)

Reference	Methodology and contributions	Pros	cons
No-regret learning: Sect. 4.3.1			
DRAGAN, Kodali et al. (2017)	Applying no-regret algorithm, new regularizer	High stability across objective functions, mitigates mode collapse	–
Chekhov GAN, Grnarova et al. (2017)	Online learning algorithm for semi concave game	Converge to mixed NE for semi shallow discriminator	–
Fictitious play: Sect. 4.3.2			
Fictitious GAN, Ge et al. (2018)	Fictitious (historical models)	Solve the oscillation behavior, solve divergence issue on some cases, applicable	Applied only on 2 player zero-sum games
Federated learning: Sect. 4.3.3			
FedGAN, Rasouli et al. (2020)	Communication-efficient distributed GAN subject to privacy constraints, connect the convergence of GAN to ODE	Prove the convergence, less communication complexity compare to general distributed GAN	–
Fan and Liu (2020)	Using a federated learning framework	Robustness to the number of clients with IID and moderately non-IID data	Performs anomaly for highly skewed data distribution, accuracy drops with non-IID data
Reinforcement learning: Sect. 4.3.4			
Florensa et al. (2018)	Generate diverse appropriate level of difficulty set of goal	–	–
Diversity-promoting GAN, Xu et al. (2018)	New objective function, generate text	Diversity and novelty	–
Zhang et al. (2018a)	Using GAN for cross-model hashing	Extract rich information from unlabeled data	–
SeqGAN, Yu et al. (2017)	Extending GANs to generate sequence of discrete tokens	Solve the problem of discrete data	–
FakeGAN, Aghakhani et al. (2018)	Text classification	–	–
CS-GAN, Li et al. (2018)	Combine RL, GAN, RNN	More realistic, faster	–
Ghosh et al. (2016a)	Handwriting recognition	–	–

Table 6 (continued)

Reference	Methodology and contributions	Pros	cons
ORGAN, Guimaraes et al. (2017)	RL agent + SeqGAN	Better result than RNN trained via MLE or SeqGAN	Works only on sequential data
MolGAN, De Cao and Kipf (2018)	RL agent + SeqGAN	optimizing non-differentiable metrics by RL & Faster training time	Susceptible to mode collapse
OptiGAN, Hossam et al. (2020)	Combining MLE and GAN	Used for different models and goals	–
Tian et al. (2020)	Redefining the issue of neural architecture search for GANs by applying Markov decision process formulation	More effective RL-based search algorithm, smoother architecture sampling	–

In Grnarova et al. (2017), Grnarova et al. used regret minimization. They provided a method that provably converges to an MN equilibrium. Because the minimax value of pure strategy for the generators is always higher than the minimax value of the mix equilibrium strategy of generators; thus, the generators are more suitable. This convergence happens for semi-shallow GAN architectures using regret minimization procedures for every player. Semi-shallow GAN architectures are architectures that the generator is any arbitrary network, and the discriminator consists of a single layer network. This method is done even though the game induced by such architectures is not convex-concave. Furthermore, they show that the minimax objective of the generator's equilibrium strategy is optimal.

### 4.3.2 Fictitious play

GAN is a two-player zero-sum game with a repeated game as the training process. If the zero-sum game is played repeatedly between two rational players, they try to increase their payoff. Let  $s_i^n \in S_i$  show the action taken by player  $i$  at time  $n$  and  $\{s_i^0, s_i^1, \dots, s_i^{n-1}\}$  are previous actions chosen by player  $i$ . So player  $j$  can choose the best response, assuming player  $i$  is choosing its strategy according to the empirical distribution of  $\{s_i^0, s_i^1, \dots, s_i^{n-1}\}$ . Thus, the expected utility is a linear combination of utilities under different pure strategies. So we can assume that each player plays the best pure response at each round. In game theory, this learning rule is called fictitious play and can help us find the Nash equilibrium. The fictitious play achieves a Nash equilibrium in two-player zero-sum games if the game's equilibrium is unique. However, if there exist multiple Nash equilibria, other initialization may yield other solutions.

By relating GAN with the two-player zero-sum game, Ge et al. designed a training algorithm to simulate the fictitious play on GAN and provide a theoretical convergence guarantee in Ge et al. (2018). They also showed that by assuming the best response at each update in Fictitious GAN, the distribution of the mixture outputs from the generators converges to the data distribution. The discriminator outputs converge to the optimum discriminator function. The authors in Ge et al. (2018) used two queues  $D$  and  $G$ , to store the historically trained models of the discriminator and the generator. They also showed that Fictitious GAN can effectively resolve some convergence issues that the standard training approach cannot resolve and can be applied on top of existing GAN variants. Furthermore, they could achieve almost 10% improvement in IS score than Durugkar et al. (2016).

### 4.3.3 Federated learning

Data limitation is a common drawback in deep learning models like GANs. We can solve this issue by using distributed data from multiple sources, but this is difficult due to some reasons like privacy concerns of users, communication efficiency and statistical heterogeneity, etc. This brings the idea of using federated learning in GANs to address these subjects (Rasouli et al. 2020; Fan and Liu 2020).

Rasouli et al. (2020), proposed a federated approach to GANs, which trains over distributed sources with non-independent-and-identically-distributed data sources. In this model, for every  $K$  time step of the local gradient, agents send their local discriminator and generator parameters to the intermediary and receive back the synchronized parameters. Due to the average communication per round per agent, FedGAN is more efficient compared

to general distributed GAN. Experiments also proved FedGAN is robust by increasing  $K$ . For proving the convergence of this model, the authors connect the convergence of GAN to convergence of an Ordinary Differential Equation (ODE) representation of the parameter updates (Mescheder et al. 2017) under equal or two time-scale updates of generators and discriminators. Rasouli et al. showed that the FedGAN ODE representation of parameters update asymptotically follows the ODE representing the parameter update of the centralized GAN. So by using the existing results for centralized GAN, FedGAN also converges.

Fan et al. also proposed a generative learning model using a federated learning framework in Fan and Liu (2020). The aim is to train a unified central GAN model with the combined generative models of each client. Fan et al. examined four kinds of synchronization strategies, synchronizing each of the central models of  $D$  and  $G$  to every client (Sync  $D$  &  $G$ ) or simply syncing the generator or the discriminator (Sync  $G$  or Sync  $D$ ) or none of them (Sync None). In situations where communication costs are high, they recommend syncing  $G$  while losing some generative potential, otherwise, synchronizing both  $D$  and  $G$ . Fan and Liu (2020) results showed that federate learning is commonly robust to the number of agents with Independent and Identically Distributed (IID) and fairly non-IID training data. However, for highly skewed data distribution, their model performed abnormality due to weight divergence.

#### 4.3.4 Reinforcement learning

Cross-modal hashing tries to map different multimedia data into a common Hamming space, realizing fast and flexible retrieval across different modalities. Cross-modal hashing has two weaknesses: (1) Depends on large-scale labeled cross-modal training data. (2) Ignores the rich information contained in a large amount of unlabeled data across different modalities. So Zhang et al. (2018b) proposed Semi-supervised Cross-modal Hashing GAN (SCH-GAN) that exploits a large amount of unlabeled data to improve hashing learning. The generator takes the correlation score predicted by the discriminator as a reward and tries to pick margin examples of one modality from unlabeled data when giving another modality query. The discriminator tries to predict the correlation between query and chosen examples of the generator using Reinforcement learning.

An agent trained using RL is only able to achieve the single task specified via its reward function. So Florensa et al. (2018) provided Goal Generative Adversarial Network (Goal GAN). This method allows an agent to automatically discover the range of tasks at the appropriate level of complexity for the agent in its environment with no prior knowledge about the environment or the tasks being performed and allows an agent to generate its own reward functions. The goal discriminator is trained to evaluate whether a goal is at the appropriate level of difficulty for the current policy. The goal generator is prepared to generate goals that meet these criteria.

GAN has limitations when the goal is for generating sequences of discrete tokens. First, it is hard to provide the gradient update from the discriminator to the generator when the outputs are discrete. Second, The discriminator can only reward an entire sequence after generation; for a partially generated sequence, it is non-trivial to balance how well it is now and how well it will be in the future as the whole sequence. Yu et al. (2017) proposed Sequence GAN (SeqGAN) and modeled the data generator as a stochastic policy in reinforcement learning (RL). The RL reward signal comes from the discriminator decided on a complete sequence and, using the Monte Carlo search, is passed back to the intermediate state-action steps. So in this method, they care about the long-term reward at every

timestep. The authors consider not only the fitness of previous tokens but also the resulted future outcome. “This is similar to playing the games such as Go or Chess, where players sometimes give up the immediate interests for the long-term victory” (Silver et al. 2016).

The main problem in Yu et al. (2017) is that the classifier’s reward cannot accurately reflect the novelty of text. So, in Xu et al. (2018) in comparison to Yu et al. (2017), Yu et al. assigned a low reward for repeatedly generated text and a high reward for “novel” and fluent text, encouraging the generator to produce diverse and informative text, and propose a novel language-model based discriminator, which can better distinguish novel text from repeated text without the saturation problem. The generator reward consists of two parts, the reward at the sentence level and that at the word level. The authors maximized the reward of real text and minimize fake text rewards to train the discriminator. The reason for minimizing the reward of generated text is that the text that is repeatedly generated by the generator can be identified by the discriminator and get a lower reward. The motivation for maximizing the reward of real-world data lies in that not only the uncommon text in the generated data can get a high reward, but also the discriminator can punish low-quality text to some extent. This method could increase the human score by 6% compared to Yu et al. (2017).

The same notion of SeqGAN can be applied in domains such as image captioning. Image captioning’s aim is to describe an image with words. Former approaches for image captioning like maximum likelihood method suffer from a so-called exposure bias problem which happens when the model tries to produce a sequence of tokens based on previous tokens. In this situation, the model may generate tokens that were never seen in training data (Bengio et al. 2015). Yan et al. (2018) used the idea of SeqGAN to address the problem of exposure bias. In this scheme, the image captioning generator is considered as the generator in the GAN framework whose aim is to describe the images. The discriminator has two duties, the first is to distinguish the real description and generated one and the second is to figure out if the description is related to the image or not. To deal with the discreteness of the generated text, the discriminator is considered as an agent which produces a reward for the generator. Although, lack of intermediate reward is another problem which solves by using the Monte Carlo roll-out strategy same as SeqGAN.

Finding new chemical compounds and generating molecules are also challenging tasks in a discrete setting. Guimaraes et al. (2017) and De Cao and Kipf (2018) tackled this problem and proposed two models that rely on SeqGAN. The main difference is adding an RL component to the basic architecture of GAN, which we discussed in Sect. 4.2.5.

The idea behind SeqGAN has also been applied to generating sentences with certain labels. Li et al. (2018) introduced CS-GAN, which consists of a generator and a descriptor (discriminator and classifier). In this model, the generator takes an action, and the descriptor task is to identify sentence categories by returning the reward. Details of this model are explained in Sect. 4.2.4.

Aghakhani et al. (2018) introduced a system that for the first time expands GANs for a text classification task, specifically, detecting deceptive reviews (FakeGAN). Previous models for text classification have limitations: (1) Biased problems like Recurrent NN, where later words in a text have more weight than earlier words. (2) Correlation with the window size like CNN. Unlike standard GAN with a single Generator and Discriminator, FakeGAN uses two discriminators and one generator. The authors modeled the generator as a stochastic policy agent in reinforcement learning (RL) and used the Monte Carlo search algorithm for the discriminators to estimate and pass the intermediate action-value as the RL reward to the generator. One of the discriminators tries to distinguish between

truthful and deceptive reviews, whereas the other tries to distinguish between fake and real reviews.

Ghosh et al. (2016a) used GANs for learning the handwriting of an entity and combine it with reinforcement learning techniques to achieve faster learning. The generator can generate words looking similar to the reference word, and the discriminator network can be used as an OCR (optical character recognition) system. The concept of reinforcement learning comes into play when letters need to be joined to form words, such as the spacing between characters and strokes from one note to another, and provide suitable rewards or penalties for the generator to learn the handwriting with greater accuracy.

The optimized generation of sequences with particular desired goals is challenging in sequence generation tasks. Most of the current work mainly learns to generate outputs that are close to the real distribution. However, in many applications, we need to generate data similar to real ones and have specific properties or attributes. Hossam et al. (2020) introduced the first GAN-controlled generative model for sequences that address the diversity issue in a principled approach. The authors combine GAN and RL policy learning benefits while avoiding mode-collapse and high variance drawbacks. The authors show that if only pure RL is applied with the GAN-based objective, the realistic quality of the output might be sacrificed for the cause of achieving a higher reward. For example, in the text-generation case, by generating sentences in which few words are repeated all the time, the model could achieve a similar quality score. Hence, combining a GAN-based objective with RL promotes the optimization process of RL to stay close to the actual data distribution. This model can be used for any GAN model to enable it to optimize the desired goal according to the given task directly.

A novel RL-based neural architecture search (NAS) methodology is proposed for GANs in Tian et al. (2020) by Tian et al. Markov decision process formulation is applied to redefine the issue of neural architecture search for GANs in this article, therefore a more effective RL-based search algorithm with more global optimization is achieved. Additionally, data efficiency can be improved due to better facilitation of off-policy RL training by this formulation (Tian et al. 2020). On-policy RL is used in most of the formerly proposed search methods employed in RL-based GAN architecture, which may have a significantly long training time because of limited data efficiency. Agents in off-policy RL algorithms are enabled to learn more accurately as these algorithms use past experience. This approach improves FID by 57% in comparison to Hoang et al. (2018). However, using off-policy data can lead to unstable policy network training because these training samples are systematically different from the on-policy ones (Tian et al. 2020). A new formulation in Tian et al. (2020) supports the off-policy strategy better and lessens the instability problem.

## 5 Open problems and future work

We conducted this review of recent progresses in GANs using game theory which can serve as a reference for future research. Comparing this survey to the other reviews in the literature, and considering the many published works which deal with GAN challenges, we emphasize on the theory aspects of GAN and not GAN applications. This is all done through the taking of a game theory perspective based on our proposed taxonomy.

Although there are various studies that have explored different aspects of GANs, but, several challenges still remain should be investigated. In this section, we discuss such

challenges, especially in the discussed subject, game of GANs, and propose future research directions to tackle these problems.

While GANs achieve the state-of-the-art performance and compelling results on various generative tasks, but, these results come at some challenges, especially difficulty in the training of GANs. Training procedure suffers from instability problems. While reaching to Nash equilibrium, generator and discriminator are trying to minimize their own cost function, regardless of the other one. This can cause the problem of non-convergence and instability because of minimizing one cost can lead to maximizing the other one's cost. Another main problem of GANs which needs to be addressed is mode collapse. This problem becomes more critical for unbalanced data sets or when the number of classes is high. In other hand, when discriminator works properly in distinguishing samples, generators gradients vanishes. This problem which is called vanishing gradient should also be considered. Compared with other generative models, the evaluation of GANs is more difficult. This is partially due to the lack of appropriate metrics. Most evaluation metrics are qualitative rather than being quantitative. Qualitative metrics such as human examination of samples, is an arduous task and depends on the subject.

More specifically, as the authors in Cao et al. (2018) expressed one of the most important future direction is to improve theoretical aspects of GANs to solve problems such as model collapse, non-convergence, and training difficulties. Although there have many works on the theory aspects, most of the current training strategies are based on the optimization theory, whose scope is restricted to local convergence due to the non-convexity, and the utilization of game theory techniques is still in its infancy. At present, the game theory variant GANs are limited, and much of them are highly restrictive, and are rarely directly applicable. Hence, there is much room for research in game-based GANs which are involving other game models.

From the convergence viewpoint, most of the current training methods converge to a local Nash equilibrium, which can be far from an actual and global NE. While there is vast literature on the GAN' training, only few researches such as Hsieh et al. (2019) formulation the training procedure from the mixed NE perspective, and investigation for mixed NE of GANs should be examined in more depth. On the other hand, existence of an equilibrium does not imply that it can be easily find by a simple algorithm. In particular, training GANs requires finding Nash equilibria in non-convex games, and computing the equilibria in these game is computationally hard. In the future, we are expected to see more solutions tries to make GAN training more stable and converge to actual NE.

Multi-agent models such as Zhang et al. (2018c), Ghosh et al. (2016b, 2018), Hoang et al. (2018), Durugkar et al. (2016), Jin et al. (2020), Hardy et al. (2019), Aghakhani et al. (2018), Mordido et al. (2020), Nguyen et al. (2017), Arora et al. (2017), Rasouli et al. (2020) and Ke and Liu (2020) are computationally more complex and expensive than two-player models and this factor should be taken into account in development of such variants. Moreover, in multi-generator structures the divergence among the generators should be considered such that all of them do not generate the same samples.

One of the other directions that we expect to witness the innovation in the future is integrating GANs with other learning methods. There are a variety of methods in multi-agent learning literature which should be explored as they may be useful when applying in the multi-agent GANs. In addition, it looks like much more research on the relationship and combination between GANs and current applied learning approach such as RL is still required, and it also will be a promising research direction in the next few years. Moreover, GAN is proposed as unsupervised learning, but adding a certain number of labels, specially in practical applications, can substantially improve its generating capability.



Therefore, how to combine GAN and semi-supervised learning is also one of the potential future research topics.

As the final note, GAN is a relatively novel and new model with significant recent progress, so the landscape of possible applications remains open for exploration. The advancements in solving the above challenges can be decisive for GANs to be more employed in real scenarios.

## References

- Aghakhani H, Machiry A, Nilizadeh S, Kruegel C, Vigna G (2018) Detecting deceptive reviews using generative adversarial networks. In: Proceedings IEEE security and privacy workshops (SPW), San Francisco, pp 89–95
- Agnes J, Herrera J, Tao H, Zhu X (2020) A survey and taxonomy of adversarial neural networks for text-to-image synthesis. *Wiley Interdiscipl Rev Data Min Knowl Discov* 10(4):1345
- Alqahtani H, Kavakli-Thorne M, Kumar G (2021) Applications of generative adversarial networks (GANs): an updated review. *Arch Comput Methods Eng* 28(3):525–552
- Arora S, Ge R, Liang Y, Ma T, Zhang Y (2017) Generalization and equilibrium in generative adversarial nets (GANs). <http://arxiv.org/abs/1703.00573>
- Bengio S, Vinyals O, Jaitly N, Shazeer N (2015) Scheduled sampling for sequence prediction with recurrent neural networks. In: Proceedings 28th International Conference on Neural Information Processing Systems (NIPS), pp 1171–1179. MIT Press, Cambridge
- Bissoto A, Valle E, Avila S (2019) The six fronts of the generative adversarial networks. <http://arxiv.org/abs/1910.13076>
- Borji A (2019) Pros and cons of gan evaluation measures. *Comput Vis Image Understand* 179:41–65
- Brocas I, Carrillo JD, Sachdeva A (2018) The path to equilibrium in sequential and simultaneous games: a mousetracking study. *J Econ Theory* 178:246–274
- Cai Y, Daskalakis C (2011) On minmax theorems for multiplayer games. In: Proceedings of the twenty-second annual ACM-SIAM symposium on discrete algorithms, pp 217–234
- Cao Y-J, Jia L-L, Chen Y-X, Lin N, Yang C, Zhang B, Liu Z, Li X-X, Dai H-H (2018) Recent advances of generative adversarial networks in computer vision. *IEEE Access* 7:14985–15006
- Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA (2018) Generative adversarial networks: an overview. *IEEE Signal Process Mag* 35(1):53–65
- Dam T, Ferdaus MM, Pratama M, Anavatti SG, Jayavelu S, Abbass H (2022) Latent preserving generative adversarial network for imbalance classification. In: 2022 IEEE International Conference on Image Processing (ICIP), pp 3712–3716
- De Cao N, Kipf T (2018) MolGAN: an implicit generative model for small molecular graphs. <http://arxiv.org/abs/1805.11973>
- Di Mattia F, Galeone P, De Simoni M, Ghelfi E (2019) A survey on GANs for anomaly detection. <http://arxiv.org/abs/1906.11632>
- Durugkar I, Gemp I, Mahadevan S (2016) Generative multi-adversarial networks. <http://arxiv.org/abs/1611.01673>
- Ebrahimi S, Meier F, Calandra R, Darrell T, Rohrbach M (2020) Adversarial continual learning. In: European Conference on Computer Vision, Springer, pp 386–402
- Fan L (2020) A survey of differentially private generative adversarial networks. In: The AAAI Workshop on Privacy-Preserving Artificial Intelligence, New York
- Fan C, Liu P (2020) Federated generative adversarial learning. <http://arxiv.org/abs/2005.03793>
- Farnia F, Ozdaglar A (2020) GANs may have no Nash equilibria. <http://arxiv.org/abs/2002.09124>
- Fedus W, Rosca M, Lakshminarayanan B, Dai AM, Mohamed S, Goodfellow I (2017) Many paths to equilibrium: GANs do not need to decrease a divergence at every step. <http://arxiv.org/abs/1710.08446>
- Florensa C, Held D, Geng X, Abbeel P (2018) Automatic goal generation for reinforcement learning agents. In: Proceedings 35th International Conference on Machine Learning, vol. 80. Stockholm, pp 1515–1528
- Franci B, Grammatico S (2020) Generative adversarial networks as stochastic Nash games. <http://arxiv.org/abs/2010.10013>
- Gao N, Xue H, Shao W, Zhao S, Qin KK, Prabowo A, Rahaman MS, Salim FD (2020) Generative adversarial networks for spatio-temporal data: a survey. <http://arxiv.org/abs/2008.08903>

- Georges-Filteau J, Cirillo E (2020) Synthetic observational health data with GANs: from slow adoption to a boom in medical research and ultimately digital twins?
- Ge H, Xia Y, Chen X, Berry R, Wu Y (2018) Fictitious GAN: training GANs with historical models. In: Proceedings European conference on computer vision (ECCV), Munich, pp 119–134
- Ghosh A, Bhattacharya B, Chowdhury SBR (2016a) Handwriting profiling using generative adversarial networks. <http://arxiv.org/abs/1611.08789>
- Ghosh A, Kulharia V, Namboodiri V (2016b) Message passing multi-agent GANs. <http://arxiv.org/abs/1612.01294>
- Ghosh A, Kulharia V, Namboodiri VP, Torr PH, Dokania PK (2018) Multi-agent diverse generative adversarial networks. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, pp 8513–8521
- Ghosh B, Dutta IK, Totaro M, Bayoumi M (2020) A survey on the progression and performance of generative adversarial networks. In: Proceedings of 11th IEEE international conference on computing, communication and networking technologies (ICCCNT), Kharagpur, pp 1–8
- Gong L, Zhou Y (2019) A review: generative adversarial networks. In: Proceedings 14th IEEE conference on industrial electronics and applications (ICIEA), Xi'an, pp 505–510
- Goodfellow I (2016) NIPS 2016 tutorial: generative adversarial networks. <http://arxiv.org/abs/1701.00160>
- Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014a) Generative adversarial nets. In: Proceedings 27th International Conference on Neural Information Processing Systems (NIPS), pp 2672–2680. MIT Press, Cambridge
- Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014b) Generative adversarial networks. <http://arxiv.org/abs/1406.2661>
- Gnarnova P, Levy KY, Lucchi A, Hofmann T, Krause A (2017) An online learning approach to generative adversarial networks. <http://arxiv.org/abs/1706.03269>
- Guimaraes GL, Sanchez-Lengeling B, Outeiral C, Farias PLC, Aspuru-Guzik A (2017) Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. <http://arxiv.org/abs/1705.10843>
- Gui J, Sun Z, Wen Y, Tao D, Ye J (2020) A review on generative adversarial networks: algorithms, theory, and applications. <http://arxiv.org/abs/2001.06937>
- Hardy C, Le Merrer E, Sericola B (2019) MD-GAN: Multi-discriminator generative adversarial networks for distributed datasets. In: Proceedings IEEE International Parallel and Distributed Processing Symposium (IPDPS), Rio de Janeiro, pp 866–877
- Hitawala S (2018) Comparative study on generative adversarial networks. <http://arxiv.org/abs/1801.04271>
- Hoang Q, Nguyen TD, Le T, Phung D (2018) MGAN: training generative adversarial nets with multiple generators. In: Proceedings International Conference on Learning Representations, Vancouver
- Hong Y, Hwang U, Yoo J, Yoon S (2019) How generative adversarial networks and their variants work: an overview. *ACM Comput Surv (CSUR)* 52(1):1–43
- Hossain M, Le T, Huynh V, Papisimeon M, Phung D (2020) OptiGAN: Generative adversarial networks for goal optimized sequence generation. <http://arxiv.org/abs/2004.07534>
- Hsieh Y-P, Liu C, Cevher V (2019) Finding mixed Nash equilibria of generative adversarial networks. In: Proceedings of International Conference on Machine Learning, Long Beach, pp 2810–2819
- Jabbar A, Li X, Omar B (2020) A survey on generative adversarial networks: variants, applications, and training. <http://arxiv.org/abs/2006.05132>
- Jain P, Jayaswal T (2020) Generative adversarial training and its utilization for text to image generation: a survey and analysis. *J Crit Rev* 7(8):1455–1463
- Jin Y, Wang Y, Long M, Wang J, Philip SY, Sun J (2020) A multi-player minimax game for generative adversarial networks. In: Proceedings IEEE International Conference on Multimedia and Expo (ICME), London, pp 1–6
- Ke S, Liu W (2020) Consistency of multiagent distributed generative adversarial networks. *IEEE Trans Cybern* 1:1–11
- Kodali N, Abernethy J, Hays J, Kira Z (2017) On convergence and stability of GANs. <http://arxiv.org/abs/1705.07215>
- Kumar MP, Jayagopal P (2021) Generative adversarial networks: a survey on applications and challenges. *Int J Multimed Inf Retrieval* 10(1):1–24
- Lee M, Seok J (2020) Regularization methods for generative adversarial networks: an overview of recent studies. <http://arxiv.org/abs/2005.09165>
- Li Y, Pan Q, Wang S, Yang T, Cambria E (2018) A generative model for category text generation. *Inf Sci* 450:301–315
- Liang J, Tang W (2020) Sequence generative adversarial networks for wind power scenario generation. *IEEE J Sel Areas Commun* 38(1):110–118

- Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2015) Continuous control with deep reinforcement learning. <http://arxiv.org/abs/1509.02971> (2015)
- Lucic M, Kurach K, Michalski M, Gelly S, Bousquet O (2017) Are GANs created equal? A large-scale study. <http://arxiv.org/abs/1711.10337>
- Lv Y, Ren X (2018) Approximate nash solutions for multiplayer mixed-zero-sum game with reinforcement learning. *IEEE Trans Syst Man Cybern Syst* 49(12):2739–2750
- Mariani G, Scheidegger F, Istrate R, Bekas C, Malossi C (2018) Bagan: Data augmentation with balancing gan. <http://arxiv.org/abs/1803.09655>
- Mescheder L, Nowozin S, Geiger A (2017) The numerics of GANs. In: *Proceedings conference neural information processing systems*, vol 30. Curran Associates Inc., Red Hook, pp 1825–1835
- Mordido G, Yang H, Meinel C (2020) microbatchGAN: Stimulating diversity with multi-adversarial discrimination. In: *Proceedings IEEE winter conference on applications of computer vision (WACV), Snowmass*, pp 3050–3059
- Mullick SS, Datta S, Das S (2019) Generative adversarial minority oversampling. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 1695–1704
- Neumann JV (1928) Zur theorie der gesellschaftsspiele. *Math Ann* 100(1):295–320
- Nguyen T, Le T, Vu H, Phung D (2017) Dual discriminator generative adversarial nets. In: *Proceedings advances in neural information processing systems (NIPS)*, pp 2670–2680
- Osborne MJ (2004) An introduction to game theory, vol 3. Oxford University Press, New York
- Pan Z, Yu W, Yi X, Khan A, Yuan F, Zheng Y (2019) Recent progress on generative adversarial networks (GANs): a survey. *IEEE Access* 7:36322–36333
- Pan Z, Yu W, Wang B, Xie H, Sheng VS, Lei J, Kwong S (2020) Loss functions of generative adversarial networks (GANs): opportunities and challenges. *IEEE Trans Emerg Top Comput Intell* 4:4
- Rasouli M, Sun T, Rajagopal R (2020) FedGAN: Federated generative adversarial networks for distributed data. <http://arxiv.org/abs/2006.07228>
- Salehi P, Chalechale A, Taghizadeh M (2020) Generative adversarial networks (GANs): an overview of theoretical model, evaluation metrics, and recent developments. <http://arxiv.org/abs/2005.13178>
- Sampath V, Maurtua I, Martín JJA, Gutierrez A (2021) A survey on generative adversarial networks for imbalance problems in computer vision tasks. *J Big Data* 8(1):27
- Sarmad M, Lee HJ, Kim YM (2019) RL-GAN-Net: a reinforcement learning agent controlled GAN network for real-time point cloud shape completion. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Long Beach*, pp 5898–5907
- Saxena D, Cao J (2020) Generative adversarial networks (GANs): challenges, solutions, and future directions. <http://arxiv.org/abs/2005.00065>
- Shin S, Jeon H, Cho C, Yoon S, Kim T (2020) User mobility synthesis based on generative adversarial networks: a survey. In: *Proceedings 22nd international conference on advanced communication technology (ICACT)*, Phoenix Park, pp 94–103
- Shoham Y, Leyton-Brown K (2008) Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge University Press, Cambridge
- Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489
- Sorin V, Barash Y, Konen E, Klang E (2020) Creating artificial images for radiology applications using generative adversarial networks (GANs): a systematic review. *Acad Radiol* 27:1175–1185
- Tian Y, Wang Q, Huang Z, Li W, Dai D, Yang M, Wang J, Fink O (2020) Off-policy reinforcement learning for efficient and effective GAN architecture search. In: *Proceedings European conference on computer vision (ECCV)*, Springer, pp 175–192
- Tran N-T, Tran V-H, Nguyen B-N, Yang L, Cheung N-MM (2019) Self-supervised GAN: analysis and improvement with multi-class minimax game. *Adv Neural Inf Process Syst (NeurIPS)* 32:13253–13264
- Tschuchnig ME, Oostingh GJ, Gadermayr M (2020) Generative adversarial networks in digital pathology: a survey on trends and future potential. *Patterns* 1(6):100089
- Wang K, Gou C, Duan Y, Lin Y, Zheng X, Wang F-Y (2017) Generative adversarial networks: introduction and outlook. *IEEE/CAA J Autom Sin* 4(4):588–598
- Wang Z, She Q, Ward TE (2019) Generative adversarial networks in computer vision: A survey and taxonomy. <http://arxiv.org/abs/1906.01529>
- Wang L, Chen W, Yang W, Bi F, Yu FR (2020) A state-of-the-art review on image synthesis with generative adversarial networks. *IEEE Access* 8:63514–63537
- Weininger D (1998) Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Comput Sci* 28(1):31–36

- Wiatrak M, Albrecht SV (2019) Stabilizing generative adversarial network training: a survey. [http://arxiv.org/abs/1910.00927](https://arxiv.org/abs/1910.00927)
- Williams RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach Learn* 8(3):229–256
- Wu X, Xu K, Hall P (2017) A survey of image synthesis and editing with generative adversarial networks. *Tsinghua Sci Technol* 22(6):660–674
- Xu J, Ren X, Lin J, Sun X (2018) Diversity-promoting GAN: a cross-entropy based generative adversarial network for diversified text generation. In: *Proceedings Conference on Empirical Methods in Natural Language Processing*, Brussels, pp 3940–3949 (2018)
- Yan S, Wu F, Smith JS, Lu W, Zhang B (2018) Image captioning using adversarial networks and reinforcement learning. In: *Proceedings 24th International Conference on Pattern Recognition (ICPR)*, Beijing, pp 248–253
- Yi X, Walia E, Babyn P (2019) Generative adversarial network in medical imaging: a review. *Med Image Anal (MEDIA)* 58:101552
- Yinka-Banjo C, Ugot O-A (2020) A review of generative adversarial networks and its application in cybersecurity. *Artif Intell Rev* 53(3):1721–1736
- Yu L, Zhang W, Wang J, Yu Y (2017) SeqGAN: Sequence generative adversarial nets with policy gradient. In: *Proceedings Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, pp 2852–2858
- Zhang S-F, Zhai J-H, Luo D-S, Zhan Y, Chen J-F (2018a) Recent advance on generative adversarial networks. In: *Proceedings of international conference on machine learning and cybernetics (ICMLC)*, Chengdu, pp 69–74
- Zhang J, Peng Y, Yuan M (2018b) SCH-GAN: semi-supervised cross-modal hashing by generative adversarial network. *IEEE Trans Cybern* 50(2):489–502
- Zhang H, Xu S, Jiao J, Xie P, Salakhutdinov R, Xing EP (2018c) Stackelberg GAN: Towards provable min-max equilibrium via multi-generator architectures. [http://arxiv.org/abs/1811.08010](https://arxiv.org/abs/1811.08010)
- Zhao Y, Ding H, Huang H, Cheung N-M (2022) A closer look at few-shot image generation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 9140–9150

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

**Monireh Mohebbi Moghaddam<sup>1,3</sup> · Bahar Boroomand<sup>1,4</sup> · Mohammad Jalali<sup>1</sup> · Arman Zareian<sup>1</sup> · Alireza Daeijavad<sup>1,5</sup> · Mohammad Hossein Manshaei<sup>1,2</sup> · Marwan Krunz<sup>2</sup>**

Monireh Mohebbi Moghaddam  
mohebi.mn@iums.ac.ir

Bahar Boroomand  
boroomand.bahar@yahoo.co.uk

Mohammad Jalali  
mjalali@ec.iut.ac.ir

Arman Zareian  
armanzareian.az@gmail.com

Alireza Daeijavad  
daeijavad@ec.iut.ac.ir

Marwan Krunz  
krunz@arizona.edu

- <sup>1</sup> Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran
- <sup>2</sup> Department of Electrical and Computer Engineering, University of Arizona, Tucson, USA
- <sup>3</sup> Iran University of Medical Sciences, Tehran, Iran
- <sup>4</sup> Department of Computing Science, University of Alberta, Edmonton, Canada
- <sup>5</sup> Department of Computing and Software, McMaster University, Hamilton, Canada