Review article

A survey on GANs for computer vision: Recent research, analysis and taxonomy

Guillermo Iglesias, Edgar Talavera *, Alberto Díaz-Álvarez

Departamento de Sistemas Informáticos – Universidad Politécnica de Madrid ETSISI - Campus Sur, C/Alan Turing, s/n, 28031, Madrid, Spain

ARTICLE INFO

ABSTRACT

In the last few years, there have been several revolutions in the field of deep learning, mainly headlined by the large impact of Generative Adversarial Networks (GANs). GANs not only provide an unique architecture when defining their models, but also generate incredible results which have had a direct impact on society. Due to the significant improvements and new areas of research that GANs have brought, the community is constantly coming up with new researches that make it almost impossible to keep up with the times. Our survey aims to provide a general overview of GANs, showing the latest architectures, optimizations of the loss functions, validation metrics and application areas of the most widely recognized variants. The efficiency of the different variants of the model architecture will be evaluated, as well as showing the best application area; as a vital part of the process, the different metrics for evaluating the performance of GANs and the frequently used loss functions will be analyzed. The final objective of this survey is to provide a summary of the evolution and performance of the GANs which are having better results to guide future researchers in the field.

Contents

* Corresponding author.
   E-mail addresses: guillermo.iglesias@upm.es (G. Iglesias), e.talavera@upm.es (E. Talavera), alberto.diaz@upm.es (A. Díaz-Álvarez).

1. Introduction

Generative Adversarial Networks (GANs) are specific Artificial Neural Networks (ANNs) architectures that were introduced in 2014 by Ian GoodFellow [1]. Generative Adversarial Networks (GANs) are a type of generative models based on game theory where ANNs are used to mimic a data distribution. Since they were firstly introduced, GANs have supposed a large change in the synthesized data generated by Artificial Intelligence (AI).

Due to their success, the number of GAN related researches has increased exponentially [2]. These researches have focused on different aspects of the models, from optimizing their training [3,4] to applying GAN to new fields such as language generation [5], image generation [3,6], image-to-image translation [7,8], image generation in text description [9], video generation [10], and other domains [11] achieving state-of-the-art results.

GAN models are capable of replicating a data distribution and generating synthesized data, applying a certain standard deviation to create new and never seen before data. Due to the particularities of GANs, one of the fields were they have supposed a change in the quality of the synthesized data is in computer vision. Although there were previous models [12–14], GANs have shown to generate sharper results [15].

The main peculiarity of GANs lies in their training, where it is based on game theory, where two neural networks compete in a min–max game. Both networks must optimize their corresponding objective functions, generating a situation where two players compete for opposites objectives.

Fig. 1 shows how the GAN architecture is composed. Due to this architecture complexity, GANs suffer from instability during their training [16–18]. The instability of training in these models gives rise to problems such as mode collapse, so that researches have been made to tackle this kind of problems [19–23]. As [24] defines, mode collapse happens when the GANs model generates the same class outputs with different inputs.

Because of the considerable variety of fields in which GANs are applied [25], the variety of different GAN architectures is wide [7,8,26]. This research focuses on outlining the fields where GANs have achieved better results. We will review the different GAN architectures that exist, how they are structured, and how they are adapted to fulfill the particularities of each problem.

Although we will explain different GAN architectures, it should be noted that, when new GAN models are created, they usually
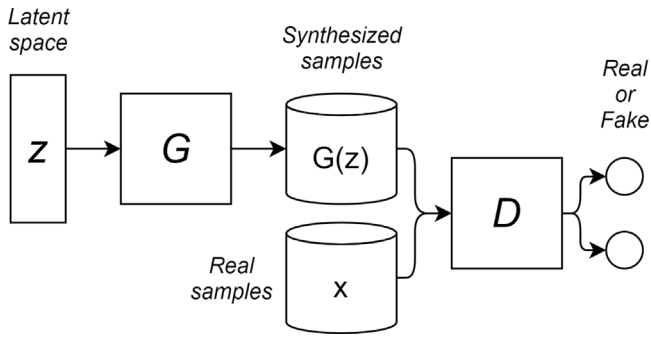
Fig. 1. Architecture of a GAN model.

combine the different results of previous researches. Most of the models that we will present are usually overlapped to achieve better results.

GAN surveys are usually focused on GAN models structure [16,27] or their application in certain tasks [28,29]. Because we will focus on novel GAN architectures, this survey can be identified as the first type. Nevertheless, in the final steps of this survey, we will review how different GAN architectures are applied to real world problems.

This survey focuses on contextualizing the recent progress in the GAN field, reviewing the different variants that have been lately presented and how they address the main problems of training GANs. We provide a complete view of the GAN structure and particularities, then we contextualize the main problems that the networks suffer. We also summarize how GAN performance is measured, explaining the most used metrics that researchers use. During the different sections we outline how the presented architectures treat the different problems that we have characterized. Finally we propose a classification of GANs based on their application, for each class we review the progress that the main variants have followed and we compare their results.

2. Related work

Several other surveys of GANs published during the last years [27,30–33] have been studied to investigate the recent trends. For example, [16] focus on the instability issues that GANs suffer and show different ways to minimize it. The results suggest that some novel architectures try to control GAN's training, while this control can be achieved by focusing on tuning hyper-parameters. It also emphasizes that much of the theoretical work does not fulfill in reality, which causes some GANs to convergence when they should not and not converge when they should.

Few surveys have been conducted to explore several approaches to optimize the loss function of GAN. This research approach tries to enhance the similarity between original and synthesized data distributions by defining an appropriate loss function. Surveys such as [34] are focus on analyzing the state-of-the-art GANs and further analyzing the performance of a huge variety of networks. In addition, they propose a set of recommendations of which loss function works best for each case of use.

Other works focusing on the applications of GANs instead of their composition or loss function. For example, [35] focus on how different GAN's architectures have been used during the last years for different problems, while [28] shows the different architectures for computer vision and their applications.

Due to the constant evolution of GANs during the last few years, these reviews are outdated almost instantaneously. As a result of some relevant and recent researches like [11,36,37]

cannot be found in any recent GAN review [25,38]. We consider that a new and more complete review must be done, covering the researches that previous reviews did not fill in and contributing to a deeper and more thorough analysis of the state-of-the-art of GANs.

3. Structure of this survey

This survey is structured as follows. Section 4 is a concise introduction of GAN composition and principles, we will also summarize the common problems that GANs suffer to then review the different solutions proposed to each problem. The different evaluation metrics are also reviewed, we address each metric strengths and weaknesses.

Section 5 reviews the most important GANs proposed since their introduction in 2014, paying special attention to the GANs proposed in the recent years. This section is divided in two types of GANs, the ones focused on improving the architecture of the GAN and the ones that tries to improve the GAN performance by changing its loss function behavior. This section also includes a new taxonomy of the reviewed articles and a timeline to have a clear vision of how the research in this field has been developed.

Section 6 summarizes the most important application of GAN architecture related to computer vision tasks. This section also includes GANs applied to other domains different than image generation, paying special attention to the treatment of different types of data, such as molecular composition or medical imaging.

Finally, Section 7 discuss the actual situation of GAN with the development of new architectures such as diffusion models or transformers. Here, we describe the potential of these new models in comparison with GAN (see Table 1).

4. Generative Adversarial Networks (GANs)

In this section, we will review the basic characteristics of GANs, their structure, composition, and common problems. We will especially focus on GAN problems because most of the GAN architectures [39,40] are created to minimize the training problems.

4.1. Definition and structure

GANs are an architecture composed of various neural networks, their objective is to replicate a data distribution in an unsupervised way. To achieve it, they are composed of two neural networks that play a two-player zero-sum game. In this game, the network called the Generator (G) is in charge of creating new data samples replicating, but not copying, the origin data distribution; while the Discriminator (D) tries to distinguish real and generated data.

From a formal point of view, D estimates $p(y|x)$, that is, the probability of a label y given the sample x; while G generates a sample given a latent space z, which can be denoted as $G(z)$.

This process consists in both networks competing. While G tries to generate more realistic results, D improves its accuracy detecting which samples are real and which not. In this process, both competitors are synchronized, if G creates a better output, it will be more difficult for D to differentiate them. On the other hand, if D is more precise, it will be more difficult for G to fool D. This process is a minimax game in which D tries to maximize the accuracy and G tries to minimize it. The formulation of the minimax game loss function can be denoted as:

$$\min_G \max_D L(D, G) = E_{x \sim p_r} log[D(x)] + E_{z \sim p_z} log[1 - D(G(x))] \quad (1)$$

where $x \sim p_r$ is the distribution of the real data and $z \sim p_z$ denotes the probability distribution of the latent space of G. $z \sim$

p_z is commonly a Gaussian or uniform noise that G uses to model new samples of data denoted as $G(z)$. D function is to differentiate between the real distribution $D(x)$ and the synthesized distribution $D(G(x))$.

According to the equation, the initial publication where the GANs where presented [1] proved the existence of a unique solution. This solution is called Nash Equilibrium (NE) and it happens when neither player can improve their loss [41].

Several researches have demonstrated that reaching the NE might not be possible in practice [42,43] or the unique solution [44].

4.2. Common problems

Due to GAN's particularities previously described, there are some aspects in GAN's training [45] to which special attention should be given.

In addition of summarizing the different main GANs problems, during Section 5 we will connect the different GAN architectures with the problems that they tackle. It should be noted that the recent proposed architectures tries to minimize the different GAN issues to optimize their models.

4.2.1. Mode collapse

The objective is to generate synthesized data from a latent space, which requires not only quality in the generated data, but generalization and diversity in the different synthesized samples. In other words, GAN models should be able to recreate new unseen data. Mode collapse occurs when the same class outputs are generated by different inputs from the latent space [46].

There are studies [21] that shows how the quality and diversity of GANs are correlated. Many efforts [20,47,48] have been taken to tackle mode collapse, but it is still an open problem.

In practice, it is not common for GAN's model to generate always the same output with different inputs [49], this issue is known as *complete mode collapse*. This type of error occurs rarely, however, it is a common problem that occurs in a partial form or partial mode collapse, in which a high number of outputs are identical. For example, in image generation, partial mode collapse happens when different outputs contain the same color or texture. It has been proven [49] that mode collapse lacks the convergence of the GANs even when NE is found.

Many of the recently proposed GAN variants tries to reduce the mode collapse problem. For example there has been proven that Wasserstein GAN (WGAN) reduces mode collapse [20,48,50].

4.2.2. Gradient vanishing

GAN's training must be balanced, both G and D need to be synchronized to learn together progressively [46,51]. A very accurate D is capable to differentiate between the real and synthesized data, this can be denoted as $D(x) = 1$ and $D(G(z)) = 0$. The loss function in this case approaches to zero, generating gradients close to zero and providing little feedback to the G. On the other hand, a poorly accurate D cannot differentiate between real and synthesized data, providing to G useless information.

4.2.3. Instability

Due to the particularity of GANs, the combination of two models learning from each other is a complex task. GAN training is based on a zero-sum game where both networks compete to find its particular solution, playing a minimax game.

This architecture of models is based on cooperation to optimize the global loss function, but the problems that D and G must optimize are opposite. Due to the particularity of the objective

Table 1
Summary of the survey.

Section		Content
4	Subsection	Generative Adversarial Networks (GANs)
	4.1	Definition and structure
	4.2	Common problems
	4.3	Evaluation metrics
5	Subsection	GAN variants
	5.1	Architecture optimization
	5.1.1	DCGAN
	5.1.2	CGAN
	5.1.3	ACGAN
	5.1.4	InfoGAN
	5.1.5	Pix2Pix
	5.1.6	CycleGAN
	5.1.7	DualGAN
	5.1.8	DiscoGAN
	5.1.9	GANILLA
	5.1.10	ProGAN
	5.1.11	DGGAN
	5.1.12	StyleGAN
	5.1.13	Alias-Free GAN
	5.1.14	SAGAN
	5.1.15	BigGAN
	5.1.16	YLGAN
	5.1.17	QuGAN
	5.1.18	EQGAN
	5.1.19	CEGAN
	5.1.20	SSD-GAN
	5.1.21	MIEGAN
	5.2	Loss function optimization
	5.2.1	WGAN
	5.2.2	WGAN GP
	5.2.3	LS-GAN
	5.2.4	lsGAN
	5.2.5	UGAN
	5.2.6	Realness GAN
	5.2.7	SNGAN
	5.2.8	CSGAN
	5.2.9	MISS GAN
	5.2.10	Sphere GAN
	5.2.11	SRGAN
	5.2.12	WSRGAN
	5.3	GAN timeline
6	Subsection	GAN applications
	6.1	Image synthesis
	6.2	Image-to-image translation
	6.3	Video generation
	6.4	Image generation from text
	6.5	Language generation
	6.6	Data augmentation
	6.7	Other domains
	6.7.1	GameGAN
	6.7.2	Medical imaging GANs
	6.7.3	GANs in agriculture
	6.7.4	Drug discovery using GANs
7		Discussion

function of the networks, there can be times during the training where a small change in one of the networks can lead to a big change in the other, in turn producing further changes. Those intervals in which both networks start to desynchronize their states are very delicate since large changes in the gradients can lead to a network losing its learning [26,52].

It should be noted that instability periods tend to generate more instability, making the problem last longer. Networks can reverse the instability process, but even if it happens, it will cost the training performance.

Many of the last proposed GAN architectures are focused on stabilize their training [3,26]. By stabilizing the training, it is usually achieved a better performance of the networks, this is why most of the last progress involve a more stable training.

4.2.4. Stopping problem

Traditional neural networks have to optimize a loss function decreasing monotonically, in theory, the cost function. Due to the minimax game that GANs have to optimize, this does not happen to them [44,53,54]. In a GAN training, the loss function does not follow any pattern, so it is not possible to know the state of the networks by their loss function. This causes that, when a training is occurring, it is not possible to know when the models have been fully optimized.

4.3. Evaluation metrics

Due to GAN's particularity, there is not an unique metric to measure the quality of the synthesized data [5]. One of the reasons of why there is no consensus among researches is the particularity of each GAN application. As mentioned in previous sections, GANs can be used to replicate any data distribution, but it depends on the particular problem how to measure the differences between the origin and synthesized distributions [55].

As there is not an unique universal metric to measure the performance of these kinds of models, during the last years there has been developed different metrics. Each metric has its particular strength and it should be noted that, in practice, different metrics are used and compared to measure different aspects and to have a wider view of the GAN performance [49].

Since there is not an evaluation metric that fulfills all GAN possible applications, we will review the most widely used metrics:

4.3.1. Inception Score (IS) and its variants

IS [45] measures the quality and diversity of the generated samples of a GAN. To do so, it uses a pretrained neural network classifier called *Inception v3* [56]. The model is pretrained using a dataset of real world images called *Imagenet* [57], it can differentiate between 1.000 of classes of images.

The IS is calculated by predicting the probabilities of the generated samples. A sample is classified strongly as one specific class means that it has high quality. In other words, it is assumed that low entropy and high quality data are correlated. The IS value varies between 1 and the number of classes of the classifier.

One of the main problems of the IS is that it cannot handle mode collapse. In this case, all generated samples by the GAN will be practically the same, but the IS would be very high if the images are strongly classified as one class. If this happens, the IS could be high and the real situation is very bad.

Other particularity of this metric is that it is designed to measure the quality of images since it uses an image classifier.

Based on IS, there are some modifications to the metric. For example, Mode Score (MS) [58] is a evaluation metric that takes into account the prior distribution of the labels over the data, i.e. it is designed to reflect the quality and diversity of the synthesized data simultaneously.

Other modification of IS is the modified-Inception Score (m-IS) [59]. It measures the diversity within the same class category output, trying to mitigate the mode collapse problem.

Some of them, like Fréchet inception distance (FID) [43] calculate the mean and covariance of the synthesized images and then calculate the distance between the real and generated image distribution. The distance is measured using the *Fréchet distance*, also known as the *Wasserstein-2 distance*. The FID is calculated as follows:

$$FID = |\mu - \mu_w|^2 + tr(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2}) \qquad (2)$$

where w denotes the synthesized data of the G.

The FID is the most common used metric to measure the quality of generated images [3,6,60,61]. The use of a common metric for different architectures allows to compare different results using a common metric. In further sections we will go through different results comparing them using FID.

One of the strengths of using this metric is that it takes into consideration contamination such as Gaussian noise, Gaussian blur, black rectangles, swirls, among others.

4.3.2. Multi-scale structural similarity for image quality (MS-SSIM)

is based on the comparison between two image structures, luminance and contrast at different scales [62]. The MS-SSIM provides a metric that compares the similarity between the real and the synthesized dataset. One of the strengths of MS-SSIM is that it correlates closer pixels with strong dependence. In comparison with other metrics such as Mean Squared Error (MSE), that calculates the absolute error of an image, MS-SSIM provides a metric based on the geometry and structure of the image.

The MS-SSIM scale is based on Structural Similarity Index Measure (SSIM), and this metric is calculated as follows:

$$SSIM(x, y) = [l_M(x, y)]^{\alpha_M}$$
$$\cdot \prod_{j=1}^{M} [c_j(x, y)]^{\beta_j} [S_j(x, y)]^{\gamma_j} \qquad (3)$$

where x and y are two windows of image of common size, l is the luminance of an image, c the contrast and S the structure. The value of SSIM is a decimal between 0 and 1, the value of 1 represents two identical sets of data. Therefore, it is assumed that the higher value of SSIM, the higher quality of the synthesized images.

MS-SSIM is calculated using the average pairwise of SSIM with N batches. This metric is commonly used with IS or its variations [63] to provide a wider view of the generated data quality.

4.3.3. Classifier Two-sample Test (C2ST)

To measure the quality of the generated distribution, a binary classifier can be used [64]. The classifiers divide the samples into synthesized and real ones, judging whether different samples belong to the same data distribution.

It should be noted that this method is not constrained to image evaluation, since a classifier can be used to classify any given data distribution, it can be adapted to any type of input data.

1-Nearest Neighbor classifier (1-NN) [65] is a type of binary classifier used to evaluate GAN performance. 1-NN is a variant of C2ST that does not require hyper-parameter tuning. C2ST using 1-NN is known as C2ST-1-NN.

Neural networks can be used as a C2ST, as mentioned in previous sections, D is indeed a classifier of real and generated data. As is proposed in [65], a C2ST can be applied to GANs by using the same composition of the discriminator, as is said in the paper "*training a fresh discriminator on a fresh set of data*". C2ST-Neural Network (C2ST-NN).

Using C2ST, we can measure the distance between the synthesized and real data distributions. This provides a useful, human-interpretable metric of GAN performance. C2ST has been applied to different GANs architectures such as DCGAN or CGAN, using C2ST-NN and C2ST-1-NN [65].

4.3.4. Perceptual path length

Using the well-known neural network classifier VGG16 [66] the perceptual path length was designed [6] to measure the entanglement of images. The embeddings of consecutive images are calculated using VGG16, interpolating random latent space inputs, then it is calculated how the synthesized images changes.

Drastic change means that, for a minimum change in the latent space there are multiple features that are changing, that means that those features are entangled under the same representation. This metric measures how well the GAN is learning the different features of the input images, measuring the entanglement of the generated images.

4.3.5. Maximum Mean Discrepancy (MMD)

is used to measure the distance between two distributions [67]. A lower score for MMD means that the distributions that are being compared are closer, and that means that the synthesized data is similar to the original.

Given distributions $\mathbb{P}$ and $\mathbb{Q}$ and a kernel k. As it is defined in [68], MMD can be denoted as:

$$
\begin{aligned}
M_k(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}^2 &= E_{\mathbb{P}}[k(x, x')] \\
&- 2E_{\mathbb{P},\mathbb{Q}}[k(x, y)] + E_{\mathbb{Q}}[k(y, y')]
\end{aligned}
\tag{4}
$$

It should be noted that this method can be used with any type of data.

4.3.6. Human rank (HR)

Human classification can be useful in some cases. Either to complement other evaluation metrics, either because there is not other metric that fulfills the particular problem, human evaluation of the generated data can be done.

Due to the particularity of this method, it can only be used when the synthesized data is comprehensive for a human.

For example, in [7,8] human classifications were applied via Amazon Mechanical Turk (AMT) to evaluate the realism of the outputs of the GAN. In this case, participants had to differentiate between the generated and real images. The more images that fool humans perception, the better.

This method can provide an approximation of how GANs creation would be perceived by humans.

5. GAN variants

Since the first GAN was developed [1] there has been published many different variations of it [3,6,8,26]. To have a broad vision about recent GAN researches, we will review the recent progress in this field.

This section is divided into GAN models according to their main features. That said, we will divide the different GAN's variations in architecture modification based and loss function modification based.

5.1. Architecture optimization

Some recent researches [3,6,7] are focused on the architecture of the GAN is designed. Some of them [3] suggest a change in GANs training, others [6] add changes to the structure of the G or D models.

Despite this, we will review traditional GAN's architecture, we will focus on models that are relevant for GAN recent development. It should be noted that the collection of architectures that we will review should not be considered individually.

GAN model evolution is supported by constant optimization. Therefore, to have a complete vision of GAN evolution, we will go through the different models that have been relevant in the last years.

5.1.1. Deep Convolutional GAN (DCGAN)

One year after, the first GAN was proposed in 2014 [1], the DCGAN was introduced [69] suggesting some changes to the original architecture. The main objective of the DCGAN is to use convolutional layers instead of the firstly proposed fully connected layers.

The main change to the fully connected GANs is the substitution of the dense layers by convolutional layers. Convolutional layers have been used during the last decade for computer vision tasks. By applying different filters to the images, the convolutional layers are able to extract the main features of the matrix of pixels keeping the correlation between adjacent pixels.

Convolutional layers are used not only used for image processing, but there are recent projects [70] that use matrices of data to take advantage of using convolutional layers.

In addition to the convolutional layers, other changes were suggested to stabilize the GAN's training. Replacing the pooling layers by strided convolution has shown better performance [71,72]. Therefore, it is proposed to use strided convolutions in both G and D.

The use of batch normalization layers in both G and D is proposed, this has been shown to reduce the noise and improve the diversity of the generated samples [73,74].

To activate the convolutional layers, it is proposed to use a Rectified Linear Unit (ReLU) activation for the hidden layer of G, hyperbolic tangent (tanh) for the output layer of G and leaky rectified linear unit Leaky Rectified Linear Unit (LeakyReLU) for D.

In addition to the mentioned changes in the architecture of the GANs, the DCGAN paper also presents a technique to visualize the filters learned by the models. This helps the comprehension of GANs learning methods, confirming previous works related to biology [75].

This architecture supposes a change in how GANs are designed and trained. The innovations that were proposed in the paper are applied in most of the following GAN models.

5.1.2. Conditional GAN (CGAN)

Proposed in 2014 [76], the CGAN architecture adds a latent class label c along with the latent space. The new label is used to split the processed data into different classes, thus the synthesized data is generated according to the class of the input label. There are some problems that require the generated data to be classified into different classes [77–79].

Despite being a simple technique, it has proven to prevent mode collapse. However, the training of a CGAN requires a labeled dataset complicating its application to some problems.

CGAN architecture has influenced GANs model since its proposition, there has been developed many variations [8,80,81].

5.1.3. Auxiliary Classifier GAN (ACGAN)

ACGAN [81] modifies the CGAN structure. The D of the ACGAN does not receive the class label c as an input, instead D is used to classify the probability of the image class. To train the model, the loss function must be modified, dividing the objective function in two parts, one for the correct source of data and the other for the class label. ACGAN loss function can be denoted as:

$$
\begin{aligned}
L_s = E[logP(S = real|X_{real})] \\
+ E[logP(S = fake|X_{fake})]
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
L_c = E[logP(C = c|X_{real})] \\
+ E[logP(C = c|X_{fake})]
\end{aligned}
\tag{6}
$$

where L_s is the log-likelihood of the correct data distribution and L_c is the log-likelihood of the correct class label.

5.1.4. Interpretable Representation Learning by Information Maximizing GANs (InfoGAN)

One of the mentioned deficiencies of conditional GANs was the requirement of a labeled dataset. InfoGAN [80] provides an architecture to train conditional GANs with an unsupervised method. To do so, the latent class label c is substituted by a *latent code* vector.

The latent space and the latent code are maximized by using the Mutual Information [82]. The mutual information term is not easy to calculate because it requires the posterior $P(c|x)$. To optimize the training performance, an auxiliary distribution

$Q(c|x)$ is defined. Said so, the loss function of the InfoGAN is defined as follows:

$$\min_{G,Q} \max_{D} V_{InfoGAN}(D, G, Q) = V(D, G)$$
$$-\lambda L_I(G, Q) \tag{7}$$

where λ is a hyperparameter that is in charge of the latent code control. As it is proposed in the original paper [80] a λ equal to 1 is used when the latent code is discrete, for continuous latent codes a smaller λ should be used. The reason for that is to control the differential entropy.

5.1.5. Image-to-Image Translation with Conditional Adversarial Nets (Pix2Pix)

The main objective of the Pix2Pix [8] architecture is to do an image-to-image translation. That is, given an image from a domain A, transform this image to other domain B. For example, given a map of a street, transform the map to an aerial photo of the street on the map.

The Pix2Pix architecture is based on an autoencoder, but skips some connections. This architecture is known as U-Net, and it is based on the idea of retrieving information at early stages of the network. The same approaches of skipping connections have been used before [83–86] showing great results and improving the network performance.

In addition to the new architecture, a new loss function is proposed that is denoted as:

$$L_{GAN}(G, D) = E_y[logD(y)]$$
$$+E_{x,z}[log(1 - D(G(x, z)))] \tag{8}$$

As a follow-up of Pix2Pix, Pix2PixHD was proposed [87] improving the quality of the generated images. Many later works have used Pix2Pix [88–91] converting it to one of the most popular architectures of the last decade.

The immediate application of these algorithms to images has had a great impact on society, radically increasing its popularity thanks to the applications developed.

5.1.6. Cycle-Consistent GAN (CycleGAN)

Cyclic consistency is the idea that, given a data x from a domain A, if the data is translated to a domain B and translated again to the A domain it should be recovered the data x. In other words, if a sample is translated to a domain and recovered from that domain, it should not change. This process, where a data sample is transformed and recovered, is known as cycle consistency, and it has been widely used during the last decades [92,93].

This idea is the main base of CycleGAN [7]. The main strength of the application of cycles is that paired data is not a requirement. GAN architecture adds a new mapping denoted as F, its function is to do the inverse mapping to retrieve the original data. In other words, the function of F is $F(G(x)) = x$. To train the architecture, a new cycle consistency loss is proposed to train the so-called *forward* and *backward* cycle consistency. The cycle consistency loss is denoted as follows:

$$L_{cycle}(G, F) = E_{x\ p_{data}(x)}[\|F(G(x)) - x\|_1]$$
$$+E_{y\ p_{data}(y)}[\|G(F(y)) - y\|_1] \tag{9}$$

Despite CycleGAN was first proposed for image-to-image translation, it can be used for any data translation.

5.1.7. Unsupervised Dual Learning for Image-to-Image Translation (DualGAN)

The architecture of DualGAN [94] is very similar to CycleGAN. As it was with the CycleGAN, the DualGAN does not require paired data to train its models. To learn the translation from one data domain to another, DualGAN has two pairs of identical G and D, each pair is responsible for their respective translation.

To stabilize the training and prevent mode collapse, the loss format of WGAN [26] is used. This marks the architecture of the network and the construction of the objective function.

In order to train each pair of G and D a *reconstruction error term* is defined. The reconstruction error objective is the same that it was in CycleGAN, calculating the distance between the original sample of data and its corresponding recovered sample. The reconstruction error is defined as:

$$l^g(u, v) = \lambda_U \|u - G_B(G_A(u, z), z')\|$$
$$+\lambda_V \|v - G_A(G_B(v, z'), z)\|$$
$$-D_B(G_B(v, z')) - D_A(G_A(u, z)) \tag{10}$$

while U and V are both domains, λ_U and λ_V are two constant parameters and z and z' are both random noises. λ_U and λ_V are normally set a value within [100.0, 1, 000.0], when the domain U contains real images (e.g. a human face photo) and V does not (e.g. a sketch of human face), it is more optimal to use a smaller value of λ_U than λ_V.

DualGAN has been widely used and modified [95–97]. For example, in [98] a DualGAN architecture was used to transform an input speech emotion. In this application, given the Fundamental Frequency (F0) of a certain emotion, the trained network is capable of changing the emotion of the sound. To do so, F0 is encoded using wavelet kernel learning [99] using the same methodology as [100].

5.1.8. Learning to Discover Cross-Domain Relations with GANs (DiscoGAN)

DiscoGAN [101] is an architecture that follows the same structure as DualGAN and CycleGAN. The particularity that DiscoGAN has is the usage of an autoencoder for the G. For D, it uses a classifier based on the encoder of the G.

Autoencoders have been used to other reconstruction problems [102–104], so applying of this architecture to domain-to-domain translation problems can benefit from their particularities. Autoencoders are based on the idea of reducing the dimensionality of the input data, then they reconstruct the same information. By doing the dimensional reduction, the network is capable of maintaining the essential features of the input data. In the case of domain-to-domain translation, by using autoencoders, the architecture is capable of maintaining the main features of a sample and translating this core information to other specific domains.

The results presented in the original work show how GANs can learn high-level relationships between two complete different domains. In the experiments carried out in the research, it was demonstrated how the networks discovered relationships such as orientation. E.g., pairing images of chairs and car with the same orientation.

5.1.9. GANILLA

The GANILLA [105] architecture modifies the structure of the G of the GAN for image style transfer. The main objective of the variant is to maintain both the content and the style of an image, previous methods usually lack one of this aspects in favor of the other. The main idea of the GANILLA is to do the style transfer of an image balancing style and content.

The architecture of GANILLA uses low-level features to maintain the content of the image at the same time as the style transfer is done. The G model is based on two stages, one for downsampling the input image and the other for upsampling the information of the first stage. This architecture ensures that the style transfer maintains the input features of the image but, in addition, some layers concatenate features of previous layers such as edges, shapes or morphological features. With these two methods, the architecture controls both content and style.
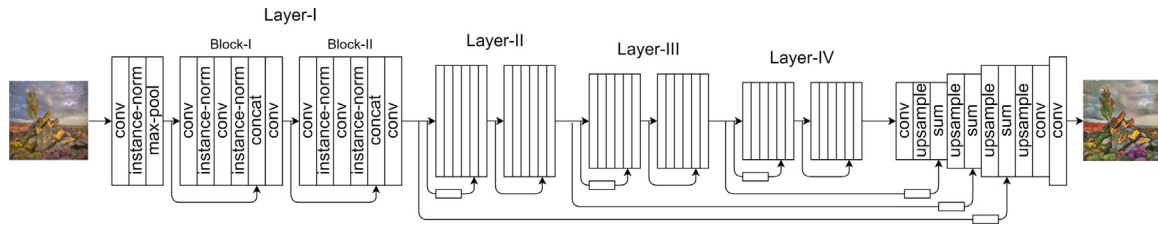
Fig. 2. Structure of the proposed architecture of the GANILLA. Figure based on Ref. [105].
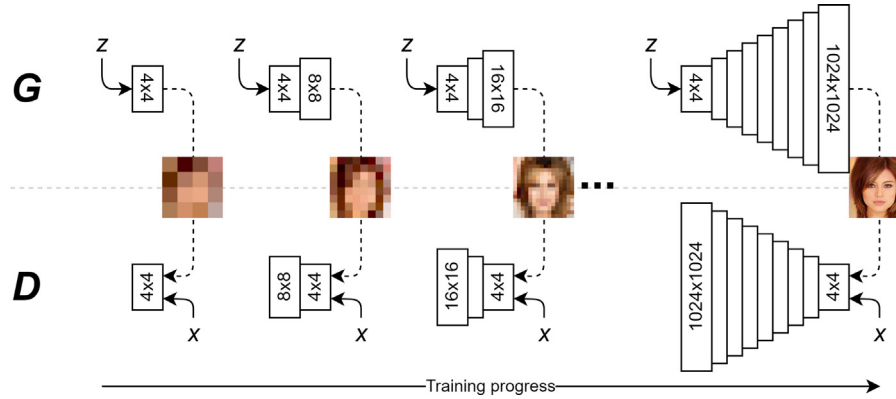

Fig. 3. Training schedule of ProGAN. Figure based on Ref. [3].

The downsampling stage is based on ResNet-18 [83] but with skipped connections. This skipped connections then feed the up-sampling module. The architecture of the GANILLA can be observed in Fig. 2

For training the models, the cyclic consistency method of the CycleGAN [7] is used. This way, two pairs of G and D are used to map both domains.

The results of the GANILLA show the good performance, in specific for children's book illustration dataset. Due to the particularities of the images of children's books, being highly contrasted images with abstract objects, previous architectures had difficulty to do the style transfer. However, with the usage of low level features of the GANILLA, it is achieved an improvement of the overall performance.

5.1.10. Progressive Growing of GANs (ProGAN)

Training a complex model can lead to strong instability. To tackle the instability of GANs models, ProGAN [3] proposes a training methodology based on a growing architecture. The idea of a progressive neural network was previously proposed [106].

The main idea behind progressive networks is the concatenation of different training phases. In each phase, a model is trained and, as the trainings are developed, the model number of layers increases. This way the created model scales up gradually stabilizing the training. The strength of this architecture is that, due to the simplicity of the first model, the networks are capable to learn properly the simplest form of the problem and then use the learned characteristics to scale up little by little the complexity of the problem. With each new phase, it is important to emphasize that the weights of the networks remain trainable, letting them to adapt to the new phases. A scheme of the progressive training of ProGAN can be seen in Fig. 3.

Due to the explained training methodology, ProGAN is capable to stabilize the training of GANs, which is one of the most important GAN problems. In addition, ProGAN's training methodology speeds up the training phase and produces images of state-of-the-art quality, e.g. achieving an inception score of 8.8 in the unsupervised CIFAR-10 [107] dataset.

The ProGAN described in the original paper used the Gradient Penalty WGAN (WGAN-GP) [4] loss format, despite that ProGAN architecture can be applied to any loss function. ProGAN training methodology has been implemented in many recent researches [108,109].

5.1.11. Dynamically Grown GAN (DGGAN)

DGGAN [110] proposes a new training methodology based on ProGAN. The architecture of the networks of DGGAN not only grow periodically, they rather grow dynamically adapting their architecture and parameters during the training.

The DGGAN questions some aspects of GANs such as the symmetry between G and D or layer choice. The new methodology can automatically search the optimal parameters, respecting ProGAN growing strategy was previously defined.

The DGGAN starts with a base D and G, the training alternates between training steps and the growing of the network. To grow the network, a set of *child* architectures are created. Each child has the same architecture as the parent, but each child proposes a different growing change to the network. During the training children architectures are trained, initializing the weights of the inherited parent layers with their respective parent weights.

In the proposed dynamic growing algorithm, each step chooses among different growing possibilities: grow G with a certain convolution layer, grow D with a certain convolution layer, or grow both G and D to a higher resolution. A scheme of the training methodology can be seen in Fig. 4.

If all children were preserved in each step, it will produce an exponential growing that would lead to large inefficiency. To avoid that, before the children generation, a prune is made. Known as greedy prune, the prune is done by keeping the top K children of each generation. Then each child becomes a parent and generates a new batch of children. The process repeats until the network grows to the desired size.

In the original research, the child search was made combining different kernel sizes and number of filters, each parameter is known as an action, and the number of total actions is denoted as T. It can be easily noted that different hyperparameters can
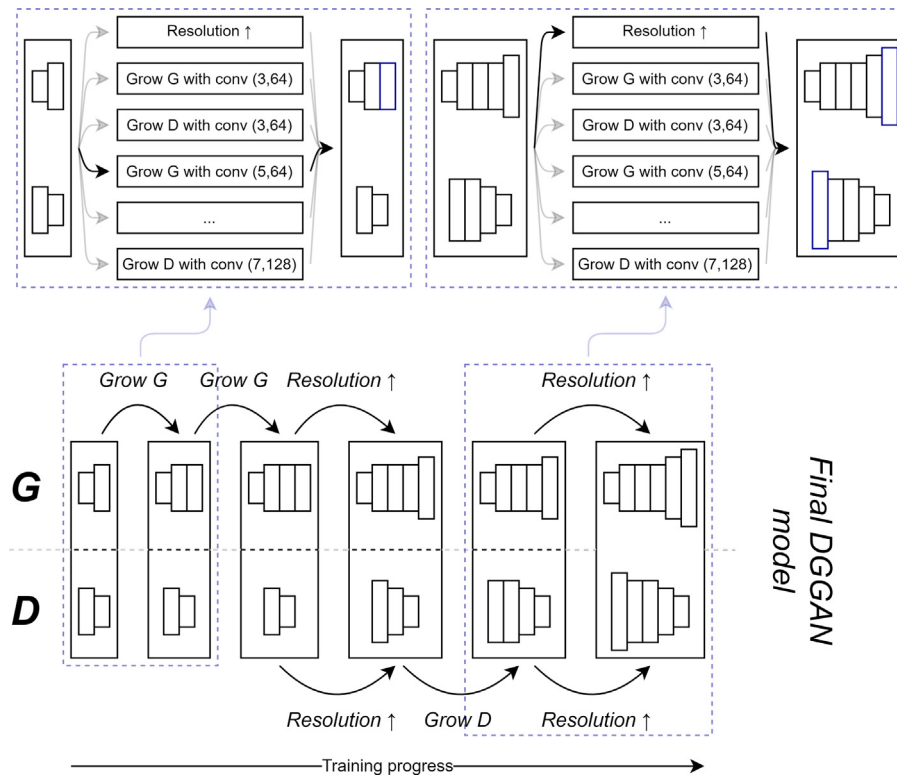
Fig. 4. Training methodology of DGGAN. Figure based on Ref. [110].

be searched by using this algorithm. To avoid a large increment of the number of children, the algorithm proposes a probability p of a child to test a new parameter. A higher K, T and p means a wider search, contributing to a better exploration of the candidates but a slower training.

It should be noted that the search algorithm lacks the efficiency of the architecture by having to do multiples training simultaneously. It also lacks the ability of growing, due to the quick growing of the number of networks.

5.1.12. A Style-Based Generator Architecture for Generative Adversarial Networks (StyleGAN)

StyleGAN [6] is based on the idea that, improving the processing of the latent space, the quality of the generated data will improve. Due to the particularities of the latent space, there are many interpolations on the variables [111,112] that produces entanglement in the learned characteristics of the G. The architecture of the StyleGAN is based on previous style transfer researches [113].

With the architecture of StyleGAN, G is capable to learn different *styles* of the input data disentangling high-level characteristics. This produces an improvement on the quality of the generated data and helps in the interpretation of the latent space, previously poorly understood. Controlling the latent space leads to better interpolation properties, enabling interpolation operations in different scales, e.g., interpolation of poses, hair or freckles in human face images.

In the StyleGAN architecture, the input of G is mapped to an intermediate latent space called W, then is used in each convolution layer via an Adaptive Instance Normalization (ADAIN). In addition to the latent space, *gaussian noise* is added to the output of each convolution layer.

The StyleGAN architecture uses the training methodology used in ProGAN, supporting the previously mentioned idea that each research should not be considered as an isolated result. The

paradigm of investigation is supported by the continuous mixing of new techniques.

Said so, the StyleGAN improves the quality of the generated images of the ProGAN, achieving a FID score of 5.06 in CelebA-HQ dataset and 4.40 in FFHQ dataset.

5.1.13. Alias-free GAN

During the last years, multiples architectures have been improving the quality of the synthesized images. The previously mentioned StyleGAN achieved one of the best results in image generation, producing images of human faces with a quality never seen before. Besides its good results, some problems remain opened.

One of the most visible problems that generated images of StyleGAN had was the known as *texture sticking*. It happens when a certain image feature depends on absolute coordinates instead depending on other feature localization. E.g. the texture of the beard of a human face seems stuck when interpolating different images. The texture sticking problem is noticeable especially when interpolating images, e.g. changing the posture of a human face image.

Alias-Free GAN [60] focus on solving the texture sticking problem of the StyleGAN. The main idea is to suppress the alias in the generated images, this way the finer details will be attached to the underlying surface of the image. To achieve this, each layer of G is designed to be equivariant by applying rotations and translations to the continuous input.

To achieve an equivariant G, many changes have been made. A 10-pixel margin is used for the internal representations, due to the assumption of infinite spatial extension for the feature maps. The Leaky ReLU layers are wrapped between an upsampling and a downsampling, this is implemented with a CUDA kernel for optimization. The cutoff frequency of the StyleGAN is cut off to ensure the alias frequencies are in the stopband. In addition, the learned input constant of StyleGAN is substituted by Fourier
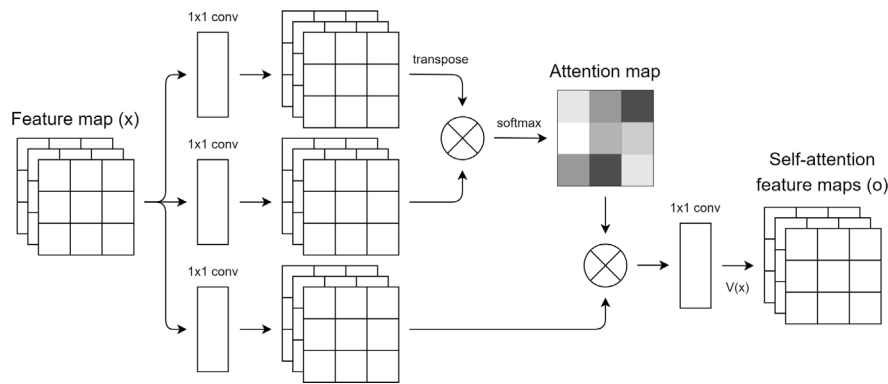
Fig. 5. Self attention layer of SAGAN. Figure based on Ref. [116].

features [114,115]. Finally, the rotation equivariant version of the network is obtained by reducing the kernel size of 3 × 3 convolutions to 1 × 1 and changing the sinc-based downsampling to a radially symmetric jinc-based one.

5.1.14. Self Attention GAN (SAGAN)

SAGAN [116] architecture covers the problem of local spatial information of images. I.e. images that have different components correlated in different positions of the image can be difficult to cover because the receptive field of the network is not big enough. In SAGAN, the generation of different features is made considering cues from all images. In addition, SAGAN D is capable of evaluating the consistency of features along the image.

SAGAN uses self attention layers [117], these layers are capable to capture structural and geometric features of multiclass datasets. The feature maps of each convolution are split into a 1 × 1 convolution in *query*, *key* and *value*, then they are multiplied to construct the output of the layer. This way the network can learn long-range dependencies. The structure of the self-attention layer can be seen in Fig. 5.

5.1.15. BigGAN

The BiGAN architecture [118] focuses on generating high resolution images from diverse datasets. Previous models results were able of synthesize new samples of low dimensionality, they had problems when scaling their results to bigger samples. The results achieved by the BigGAN, in terms of FID and IS outperform previous models.

The researches of the BigGAN claim that GANs have better performance when they use higher dimensional data. The architecture of the BigGAN is based on the SAGAN [116] architecture. The authors show that, by enlarging the number of channels of the images used by a factor of 50%, the IS improve by a factor of 21%.

One innovation proposed in this article is the so-called "Truncation Trick". Previous GAN models used a normal or uniform distribution to generate the latent space of the G network. The authors claim that by using a truncated normal distribution the results, in terms of FID and IS, were better. This truncation trick reduce the variety of values of the latent space by truncating them towards zero. The main drawback produced by this is that the variability of the generated samples is reduced. It exists a relationship between the variety and fidelity of the generated samples using this truncation. The more truncation applied to the latent space, the less variety of images were produced.

Other aspect that is scaled up in this work is the batch size of the GAN training, increasing it by a factor of 8. The authors show that by using larger batches the gradients of each iteration are better, reaching a better performance in less steps. This is caused because the composition of each batch is more diverse, being able of covering more modes of the data.

5.1.16. Your Local GAN (YLGAN)

YLGAN [61] proposes a new attention layer that substitutes the SAGAN dense attention layer [116]. This new layer preserves two-dimensional image locality and contributes the flow of information through the different layers. To preserve the two-dimensional locality and quantify how information flows through the model, the framework of Information Flow Diagram (IFD) [119] is used.

The modification of the self attention layer of SAGAN introduces sparse attention layers. This new method reduces the quadratic complexity of the attention layer by splitting the attention into multiple subsets of data. The main problem of the sparse attention layer is that, besides its computational optimization, it lacks the information flow of the network. To tackle this information flow graphs are introduced, these graphs will be used to support Full Information through the layers of the network.

The results show how applying the new layer improves the quality of the images compared to the SAGAN generated images. The architecture of the SAGAN, modifying the dense attention layer and preserving the rest parameters is called YLG-SAGAN. YLG-SAGAN not only improves the FID of SAGAN, reducing it score from 14.53 to 8.95, furthermore it optimizes the training time to around a 40%.

5.1.17. A GAN Through Quantum States (QuGAN)

During the last decade, quantum computing has become a hot topic in computer science. Since it was proposed in 1980 [121] it has always been restricted to a few laboratories around the world. Thanks to the progress made recently [122], it has made possible to test the first algorithms, prototypes and ideas [123].

Thanks to quantum computing particularities, problems previously defined can be solved, or are optimized, reducing their computation time. Using quantum superposition, the multiples solutions can be evaluated simultaneously, then by using quantum interference and entanglement the correct answer can be defined.

QuGAN [124] proposes a GAN architecture powered by quantum computing. By using quantum computing, GANs are hugely optimized, reducing a 98.5% of its parameter set compared to traditional GANs.

QuGAN architectures use qubits to create the quantum layers of G and D, known as QuG and QuD. The data that the networks use is transformed into quantum states.

5.1.18. Entangling Quantum GAN (EQGAN)

EQGAN [125] proposes a variation of the previously proposed quantum GANs. Benefiting from the entangling properties of quantum circuits, EQGANs guarantees the convergence to a NE.

The main particularity of EQGAN is that it performs quantum operations on both synthesized and real data. This approach
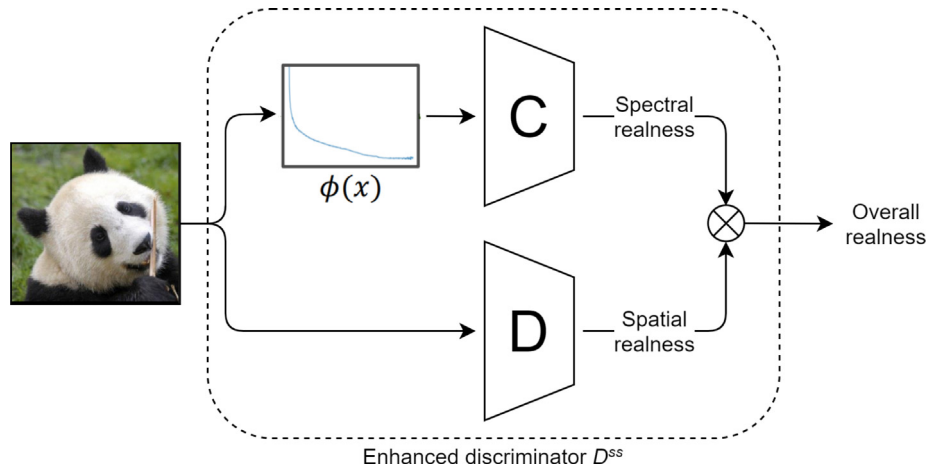
Fig. 6. Structure of the proposed enhanced D of the SSD-GAN. Figure based on Ref. [120].

produces fewer errors than swapping the data between quantum and classical.

To apply EQGAN to real problems, a Quantum Random Access Memory (QRAM) is used. By using the QRAMs, the EQGAN is capable to improve the performance of the D.

5.1.19. Classification Enhancement GAN (CEGAN)

Data imbalance is a common problem when using real world datasets. Dataset often contains a majority of samples of a certain data class. In the case of GANs using unbalanced datasets, the imbalance problem results in poor quality of the synthesized data of the class with less samples.

CEGAN [40] tries to solve the data imbalance problem in GAN. The objective is to enhance the quality of the synthesized data and to improve the accuracy of the predictions.

The CEGAN architecture consists of 3 different networks, G, D and a new network known as the classifier (C). The training of the CEGAN divides in two steps. In the first step, the architecture is normally trained, using D to differentiate between fake and real samples, C is used to classify the class label of the input sample. Then, in the second step, an augmented training dataset is formed via generating new samples from G, and this new dataset is used to train the C.

The methodology presented in CEGAN substitutes previous techniques to deal with data imbalance. Unlike other methods such as undersampling [126] or oversampling [127] CEGAN does not modify the original dataset. This way, some problems of the traditional methods are avoided, e.g. shortening the original dataset by undersampling or redundant information by oversampling with geometric transformations.

5.1.20. SSD-GAN

The SSD-GAN [120] tackles the problem of high frequency samples in GANs. The described problem causes high spectrum discrepancies between the real and the synthesized samples. The SSD-GAN proposes to alleviate this discrepancy to enhance the quality of the synthesized data.

The idea behind the architecture is to reduce the gap of spectrum discrepancy, combining the spectral realness and the spatial realness of each sample, to do so a new D is defined. The new proposed D is known as D^{ss} and it combines D and a classifier C. D is in charge of measuring the spatial realness of an image, this is the same approach of the D of the traditional GAN [1]. The new proposed C is in charge of the known as *spectral classification*, this is, measure the difference of the specters of synthesized and real

data. The C objective function is called spectral classification loss and it is defined as:

$$\mathcal{L}_{spectral} = \mathbb{E}_{x \sim p_{data}(x)}[logC(\phi(x))] \\ + \mathbb{E}_{x \sim p_g(x)}[log(1 - C(\phi(x)))] \tag{11}$$

One of the strengths of the SSD-GAN is its simplicity, easing implementation and allowing its implementation on various network architectures without excessive cost. The Fig. 6 shows how both spatial and spectral information are processed by the new D proposed for the SSD-GAN.

SSD-GAN results show the potential of the proposed architecture. The quality of the images enhances the results of previous architectures, e.g. reducing the FID score of the StyleGAN [6] from 4.40 to 4.06 by including the spectral classification.

5.1.21. Mobile Image Enhancement GAN (MIEGAN)

The MIEGAN [128] presents a novel architecture that aims to improve the quality of images taken with a mobile phone. To do so, two new networks are proposed, the so-called multi-mode cascade generative network and the adaptive multi-scale discriminative network. The generative network is composed of an Autoencoder architecture. The encoder of this new generator is divided into two streams, the inclusion of the second encoder is in charge of improving the low luminance areas, where mobile phones particularly lack in their clarity.

The discriminator network has a dual goal. First, the global discriminator ensures overall image quality. Second, a local discriminator maintains the local quality of small areas of the image. To combine both objectives, an adaptative weight allocation module is also proposed that is responsible for balancing the importance of each discriminator.

A brief scheme reviewing all presented architecture variant GANs can be seen in Fig. 7. We divide the different architecture-based GANs in different groups based on the proposed changes. The illustration gives a global view of how are interconnected different researches of the last years.

5.2. Loss function optimization

Orthogonal to the architecture modification GANs, there are many researches [18,26,129] that focuses on the objective function of GANs. For example, the instability problem of GANs is actually caused by the Jensen–Shannon divergence, where D often wins over G. Along with architecture optimization GANs, there have been developed loss optimization researches, where both approaches coexist and interact with each other.

In this section, we will review the different most important and recent progress in variations of the loss function of the GANs.
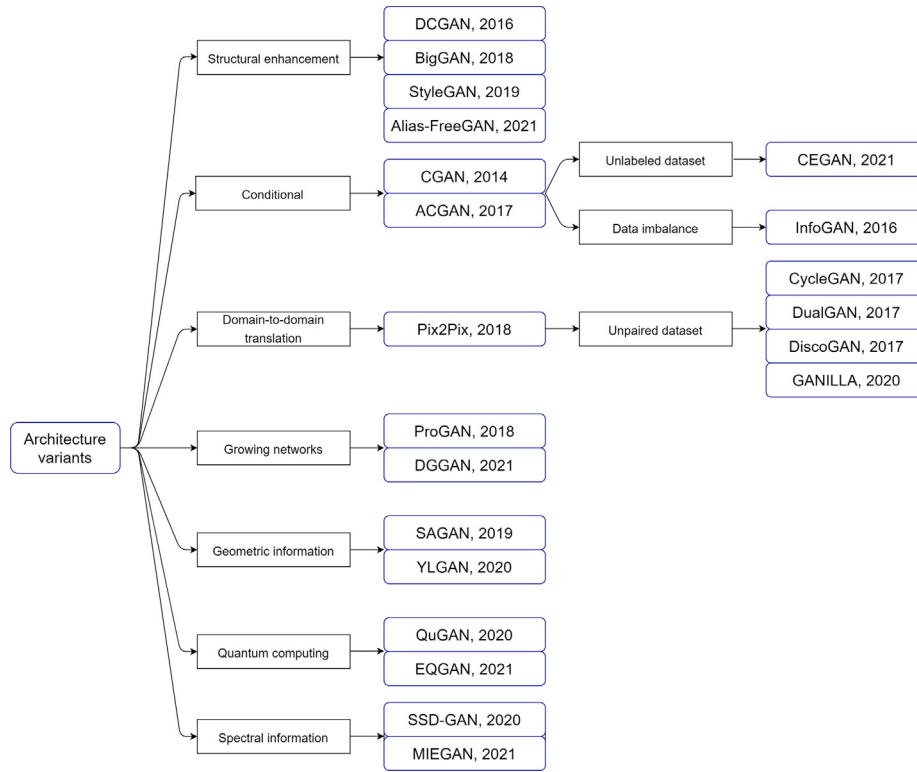
Fig. 7. Survey proposed division of architecture variants for GANs.

5.2.1. WGAN

The base of the WGAN [26] is the application of the *Earth Mover* (EM) distance, also known as *Wasserstein-1* distance. The Wasserstein distance is defined as:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|] \tag{12}$$

In other words, the Wasserstein distance calculates the cost of transforming the distribution $\mathbb{P}_r$ to the distribution $\mathbb{P}_g$. In the case of GAN, the Wasserstein distance will measure the difference between the real and synthesized data distributions.

In order to apply the new objective function, some changes must be applied to the architecture of GANs. The D of the GAN changes its objective, but previously D was used to distinguish which data was real and which was synthesized in WGAN D change its name to *critic*. The critic function is to measure the *realness* of an image, e.g. the probability that the image belongs to the real distribution. The weight change of the critic is fixed between a window (e.g. between [-0.01, 0.01]) after each gradient update. The weight clipping is done to make the parameters lie in a compact space, due to the change of the critic network.

The EM distance has shown to produce better gradient behavior than other metrics. The results of the original paper show that, compared with the classical GAN loss function, the WGAN has better behavior in terms of convergence, mode collapse avoiding and stability. Particularly in low-dimensional manifold distributions, WGAN has shown to outperform traditional JS and KL divergences [130]. Other important benefit of WGAN is that the loss correlates with the quality of the synthesized samples and converges to a minimum.

WGAN is one of the most adopted variants, due to its capacity to deal with instability and mode collapse. Many later GAN variants [6,106] use the WGAN loss function along with their own changes. For example, the Multi-marginal Wasserstein GAN (MWGAN) [131] proposes a new objective function based on WGAN for multi marginal domain translation.

5.2.2. WGAN-GP

In the original paper of WGAN, the authors suggest that weight clipping is "a terrible way to enforce Lipschitz constraints". Weigh clipping is one problem that the original WGAN had, but it worked well enough and its implementation was easy. The WGAN-GP [4] proposes a new technique to substitute the weight clipping that leads to the WGAN with undesired behavior.

The proposed change involves constraining the critic gradient norm output regarding to the input of the network. The constraint is softened via a penalty on the gradient norm. say that the new loss function is denoted as follows:

$$L = \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)]+$$
$$\lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1)^2] \tag{13}$$

The new change makes the WGAN-GP optimize its training, stabilizing it with almost no hyperparameter tuning. The new loss function also improves the quality of the generated images over WGAN and converges faster.

5.2.3. Loss-Sensitive GAN (LS-GAN)

In order to measure the quality of the synthesized samples of data created by G, a new loss function is used in the LS-GAN [129]. The new loss function aims to use regularization theory to improve the performance of GANs architecture. The main idea behind the new loss function is that a real sample produces smaller losses than a synthesized one, the margin between both is predefined. Once this assumption is set, we can infer that the training of G must aim at minimizing the loss margin between real and synthesized images. The proposed loss function is denoted as follows:

$$\min_D \mathcal{L}_D = \mathbb{E}_{x\sim p_r}L_\theta(x) + \lambda\mathbb{E}_{\substack{x\sim p_r \\ z\sim p_z}}(\Delta(x, G(z))$$
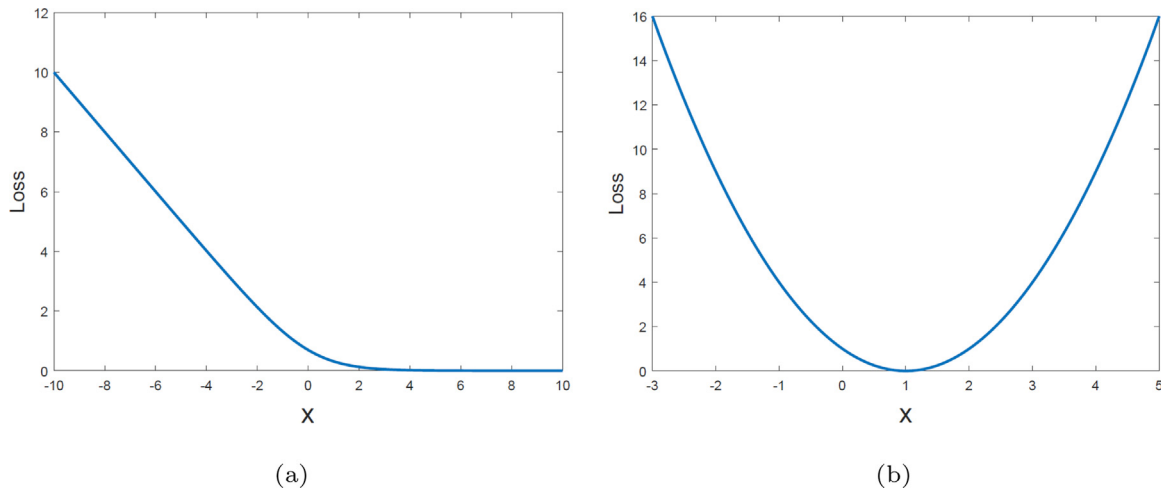$$+L_\theta(x) + L_\theta(G(z)))_+ \tag{14}$$

Fig. 8. Comparison between sigmoid cross entropy loss function (a) and least squares loss function (b).
Source: Figure from Ref. [18].

$$\min_{G} \mathcal{L}_G = \mathbb{E}_{z \sim p_z} L_\theta(G(z)) \tag{15}$$

where λ is a hyperparameter for balancing and θ are the parameters of D.

The loss function is regularized via Lipschitz regularity condition over the density of the real data. Due to the regularization, the created models are better in generalization of new data.

5.2.4. Least Square GAN (LSGAN)

The new loss function presented in LSGAN [18] aims to reduce the vanishing gradient problem. The main objective of the LSGAN is to punish the synthesized samples that are far from the real data but still in the correct side of the decision boundary. The least squares loss function is denoted as follows:

$$\min_{D} V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x \sim p_{data}(x)}[(D(x) - b)^2] \\ + \frac{1}{2}\mathbb{E}_{z \sim p_z(z)}[(D(G(z)) - a)^2] \tag{16}$$

$$\min_{G} V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z \sim p_z(z)}[(D(G(z)) - c)^2] \tag{17}$$

where a and b are the labels for fake and real data respectively and c is the label that G wants D to believe is real data. It should be noted that the square of both equations is responsible for punishing far from the decision boundary samples.

The LSGAN tries to generate more gradients while penalizing samples that lie a long way from the decision boundary. This way the gradients are forced to be higher, preventing the gradient vanishing problem. Compared to the classical sigmoid cross entropy loss function of GANs, the new least squares loss is flat only at one point as we can see in Fig. 8.

5.2.5. Unrolled GAN (UGAN)

The UGAN [39] loss function is defined to prevent instability in GANs training. The idea behind UGAN is to dynamically adapt G and D to prevent the situation of unbalance, where one of the networks is more trained than the other. Commonly, due to the particularity of the problem to solve, the D problem is easier to solve than the G one, producing an imbalance in favor of the D.

The training of UGAN is dynamically changed, the presented loss is surrogated for training the G. The surrogate objective function is created by unrolling K steps of D for each update of the G. Using the proposed loss function, the G behavior adapts to the training state of the D. The surrogate loss function is defined as follows:

$$\frac{df_K(\theta_G, \theta_D)}{d\theta_G} = \frac{\partial f(\theta_G, \theta_D^K(\theta_G, \theta_D))}{\partial \theta_G} \\ + \frac{\partial f(\theta_G, \theta_D^K(\theta_G, \theta_D))}{\partial \theta_D^K(\theta_G, \theta_D)} \frac{d\theta_D^K(\theta_G, \theta_D)}{d\theta_G} \tag{18}$$

With the application of the proposed loss function, the UGAN demonstrates to stabilize the training by adjusting and synchronizing G and D networks. Furthermore, it prevents mode collapse, avoiding the model to drop regions of the data distribution. Despite this, the most important weakness of the UGAN is its computational cost. When the generator loss is optimized, the performance of the network drops. It depends on the particular problem how many unrolls need to stabilize its training. In the original paper, for example, it varies between 1 and 10.

5.2.6. Realness GAN

The new variation presented by RealnessGAN [132] is a generalization of the original version of the GAN. The proposed loss function changes the output of D, making it a distribution of the realness of the input data. In other words, the discriminator function is to measure the potential realness of the input data. The proposed loss function is defined as follows:

$$\max_{G} \min_{D} V(G, D) = \mathbb{E}_{x \sim p_{data}}[\mathcal{D}_{KL}(\mathcal{A}_1 \parallel D(x))] + \\ \mathbb{E}_{x \sim p_g}[\mathcal{D}_{KL}(\mathcal{A}_0 \parallel D(x))] \tag{19}$$

where $\mathcal{A}_0$ and $\mathcal{A}_1$ are the fake and real distributions.

Using the new loss function, the RealnessGAN is capable of recovering more modes than a standard GAN, preventing mode collapse. Furthermore, RealnessGAN shows a better performance, generating higher quality images in both real-world and synthetic datasets.

One of the strengths of the RealnessGAN is its simple implementation, due to the fact that RealnessGAN is a generalization of the original GAN. That said, despite being one of the most recently proposed architectures, it is expected to be widely used due to its good results and easy implementation.

5.2.7. Spectral Normalization for GANs (SN-GAN)

SN-GAN [133] proposes a new technique to normalize the weights of D networks. A more stable training is searched through spectral normalization.
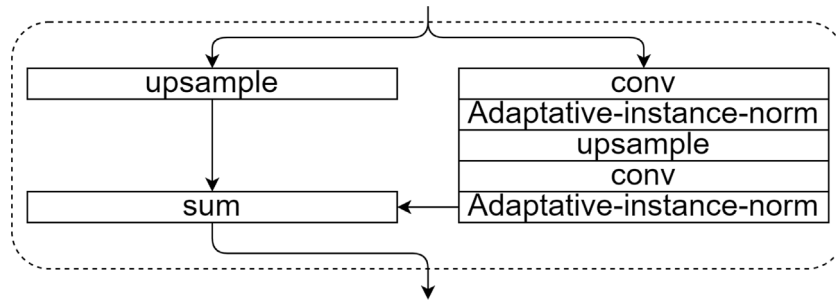
Fig. 9. Structure of the proposed residual blocks of the MISS GAN. Figure based on Ref. [139].

Respect previous normalizations [134] spectral normalization is easier to implement. The previous methods imposed a much stronger constraint on the network matrix. With the spectral normalization, it is possible to relax this constraint, allowing the network to satisfy the local 1-Lipschitz constraint. The spectral normalization is defined as follows:

$$\bar{W}_{SN} := W/\sigma(W) \tag{20}$$

where W is the weight matrix of D and $\sigma(W)$ is the L_2 normalization of W.

As mentioned before, the proposed D network is very simple and additionally its computational cost is small. It also requires the tuning of one hyperparameter, the Lipschitz constant.

The generated images using SN-GAN are more diverse, achieving better comparative IS respecting other weight normalizations.

5.2.8. Cyclic-Synthesized GAN (CSGAN)

CSGAN [135] proposes a new loss function for image-to-image translation problems. Previous works developed architectures for concrete domains of translation, CSGAN proposes a common framework for different domain translation.

The Cyclic-Synthesized Loss (CS) is proposed as the objective function of CSGAN. The new loss objective is to evaluate the differences between a synthesized image and its correspondent cycled image. The proposed loss function is denoted as follows:

$$\mathcal{L}(G_{AB}, G_{BA}, D_A, D_B) = \mathcal{L}_{LSGAN_A} + \mathcal{L}_{LSGAN_B}$$
$$+ \lambda_A \mathcal{L}_{cyc_A} + \lambda_B \mathcal{L}_{cyc_B} + \mu_A \mathcal{L}_{CS_A} + \mu_B \mathcal{L}_{CS_B} \tag{21}$$

were $\mathcal{L}_{CS_A}$ and $\mathcal{L}_{CS_B}$ are the Cyclic-Synthesized loss of both domains.

With respect to previous architectures, CSGAN produces images of better quality, notably reducing the artifacts of the synthesized images. The results show better performance of CSGAN in Chinese University of Hong Kong (CUHK) dataset [136] and comparable performance in FACADES dataset [137]. The comparison of the performance is made against GAN [1], Pix2Pix [8], DualGAN [94], CycleGAN [7] and Photo-Sketch Synthesis using Multi-Adversarial Networks (PS2MAN) [138].

5.2.9. Multi-IlluStrator Style GAN (MISS GAN)

The proposed architecture of MISS GAN [139] presents only one trained model to generate illustrations for different image styles. Previous methods used different G for each style, limiting the practical application of the architectures, while MISS GAN uses a unique model.

The proposed new G is based on the GANILLA [105] architecture, but it proposes some changes to the architecture of the decoder of the GANILLA G. The new decoder contains three residual blocks, these residual blocks are in charge of processing the low-level features from previous layers. The composition of each residual block can be seen in Fig. 9.

To train the MISS GAN models five different objective functions are proposed.

The first loss function is called the *adversarial objective* ($\mathcal{L}_{adv}$) and it is in charge of, taking the input image and the target domain, ensure that the generated image style corresponds with the target domain. To do so the $\mathcal{L}_{adv}$ takes two discriminator predictions, one for the input image and other for the synthesized image.

The second loss function is denoted as *style reconstruction objective* ($\mathcal{L}_{sty}$), and it enforces the G to use the mapping network style code while receiving a generated latent code, to calculate the $\mathcal{L}_{sty}$ the output of the G encoder over the generated image.

The third proposed objective function is called *style diversification objective* ($\mathcal{L}_{ds}$) and it compares a pair of synthesized images, each image corresponds to a different style code, each one generated from a different latent code. The objective of this loss function is to force G to produce diverse images, preventing two images with different latent codes from being the same.

The fourth objective function is the *cycle consistency loss* ($\mathcal{L}_{cyc}$) used in the CycleGAN [7].

Finally, the fifth objective function is called *content features loss* ($\mathcal{L}_{content\_feat}$), and it computes the distance in the feature space by using a VGG16 [66] network.

To combine the different objective function a *total objective* is defined as follows:

$$\max_D \min_{G,F,E} \mathcal{L}_{adv} + \lambda_{sty}\mathcal{L}_{sty} - \lambda_{ds}\mathcal{L}_{ds}$$
$$+ \lambda_{cyc}\mathcal{L}_{cyc} + \lambda_{feat}\mathcal{L}_{content\_feat} \tag{22}$$

where E is the style encoder and F is the mapping network, all the λ parameters correspond to a hyperparameter for each objective function.

5.2.10. Sphere GAN

SphereGAN [140] proposes a new architecture based on integral probability metric (IPM). The main characteristic of SphereGAN is that it bounds the IPMs objective function on a hypersphere.

Compared with other architectures such as WGAN-GP [4] SphereGAN loss function does not require any constraint term, reducing the necessity of hyperparameter tuning. The loss function of SphereGAN is defined as follows:

$$\min_G \max_D \sum_r E_x[d_s^r(N, D(x))] - \sum_r E_z[d_s^r(N, D(G(z)))] \tag{23}$$

where d_s^r denotes the r-th moment distance between a sample and the north pole of the hypersphere.

In the original paper, the mathematical properties of SphereGAN are proved, showing that minimizing the objective function of SphereGAN is equivalent to reducing IPM. In addition, it is proved that SphereGAN compared to WGAN can use r-Wasserstein distances, unlike WGAN that could only use
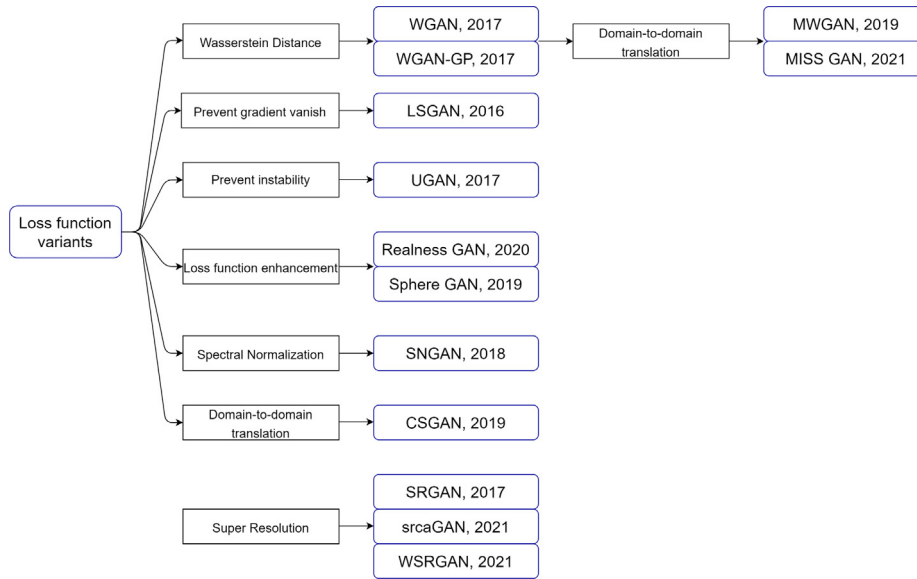
Fig. 10. Survey proposed division of loss function variants for GANs.

1-Wasserstein distance. This provides to SphereGAN a wider function space.

The SphereGAN results show its good performance, achieving a IS of 8.39 and FID score of 17.1 in CIFAR-10 [107] dataset. Compared to WGAN-GP that achieved IS of 7.86 in the same dataset.

5.2.11. Super Resolution GAN (SRGAN)

In order to apply GANs to image upscaling the SRGAN [141] was proposed. The proposed GAN objective is to take an input natural image and upscale it resolution by a factor of 4.

To achieve the super resolution, the new variant proposes a couple of adversarial and content losses. Both functions are combined using the called *perceptual loss function*, this function is in charge of a solution respecting the relevant characteristics of the data. The content loss is defined as follows:

$$l^{SR} = l_X^{SR} + 10^{-3} l_{Gen}^{SR} l_{Gen}^{SR} \tag{24}$$

where l_{Gen}^{SR} is the adversarial loss and l_X^{SR} is the content loss.

The content loss used relies on a pre-trained VGG-19 model [66]. This model, respecting the usage of a loss function such as MSE is more invariant to changes in pixel space. This metric will provide the network information about the quality of the content of the synthesized image. The new loss function is calculated as:

$$l_{VGG/i.j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} \\ - \phi_{i,j}(G_{\theta G}(I^{LR}))_{x,y})^2 \tag{25}$$

where I^{LR} refers to the low resolution images and I^{HR} refers to the high resolution image.

In addition to the content loss, the adversarial loss is defined as being this part of the generative component of the GAN. This function is responsible for pushing the generated images to be realistic and indistinguishable from the real ones. The loss function is defined as:

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -log D_{\theta D}(D_{\theta G}(I^{LR})) \tag{26}$$

The application of SRGAN improves the results of previous algorithms for image super resolution.

Since the introduction of the SRGAN it has been used in many different applications [142–144]. In addition, there are works such as [145] that presents some improvements in the SRGAN structure, the new architecture is known as Super Resolution Channel Attention GAN (srcaGAN). The architecture presented in this papers adds a channel attention module to the models, this module recovers the attention layer used in SAGAN [116]. The results presented in this new architecture outperforms the SRGAN.

5.2.12. Weighted SRGAN (WSRGAN)

One of the characteristics of the SRGAN [141] was the combination of the content loss and adversarial loss during the training. The WSRGAN proposes is changing the importance of each loss and studying the effect of this action.

The main objective of the WSRGAN is to improve the performance of the architecture by analyzing its performance in different combinations of its objective functions. Then the new weighted loss function is defined as follows:

$$l_X^{SR} = w l_{MSE}^{SR} + (1-w)10^{-3} + l_{VGG}^{SR} \tag{27}$$

where w is the parameter that controls the impact of each loss function on the final result.

After training the network with different weight configurations, the paper concludes that the MSE loss is the most important loss function, being supported by the VGG loss.

Additionally, the definition of the weight parameter is declared dynamically, obtaining even better results than when it is static.

A brief scheme reviewing the different presented loss function variant GANs can be seen in Fig. 10. We divide the different GANs in different groups based on the proposed changes in the loss function.

5.3. GAN timeline

A timeline with the reviewed architectures is presented in Fig. 11. The GANs that have been studied during Sections 5.1 and 5.2 are showed temporally. This timeline provides an overview of the historical development of GANs.

As it can be seen, the timeline compiles the most important works of the last decade. It is important to analyze that
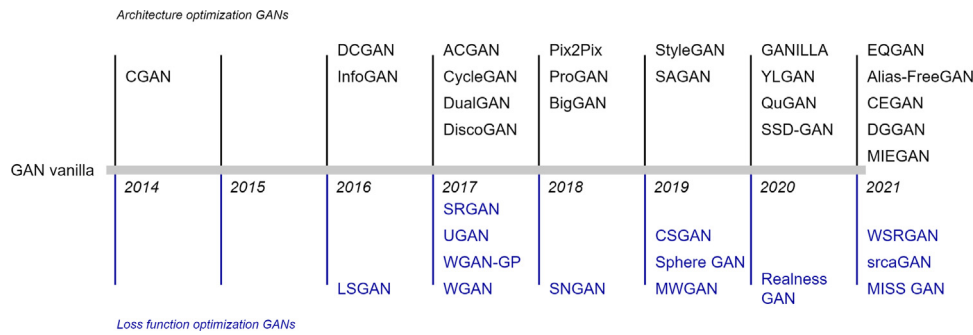
Fig. 11. Timeline of the reviewed GAN architectures.

some researches have influenced posterior ones. In some cases some researches adopt innovation of previous works as a base to then propose new changes, e.g. the DCGAN that have influenced several posterior works. In other cases there are relationships between works can be seen as an unique research, linking each article with each other by taking previous results and improving them, e.g. in the case of ProGAN, StyleGAN and Alias-Free GAN.

6. GAN applications

As mentioned before, GANs are one of the most popular applications of machine learning of the last years. GANs models can achieve results in fields where previous models could not, in other cases, GANs improve the previous results significantly.

In this section, we will review the most important fields where GAN architectures are applied, paying a special attention to the GAN models related to computer vision tasks and we will compare the different architecture results.

Most of the last researches focus on how to apply GANs to generate new synthesized data, replicating a data distribution. But, as we will review in this section GANs can be applied to other fields, e.g. video game creation [11].

6.1. Image synthesis

One of the most important fields in which GANs are applied is in computer vision. In particular, realistic image generation is the most widely used application of GANs [3,6,26].

Most of the proposed GAN variants are tested by generating real world images. Arguably, image synthesis is the first application one might think of when thinking about GAN. Its popularity is due to the good results that GAN can achieve. Compared with previous methods, GANs provide sharper results [146]. Both in academic world and for the general public GAN has raised a lot of interest.

One of the main reasons of the GAN success is its results easy understanding. As the mainly generated output of GANs are images, they can be easily understood by anyone. Even if a person does not have any technical understanding of artificial intelligence, it is possible to judge the results.

Within computer vision, image generation is the most used method to test GANs. There are plenty of real world images datasets that can be used to train GANs. The availability of datasets that can be used for training neural networks is usually the main drawback of artificial intelligence projects. Either by its availability or by its content [147] having a good dataset is essential for machine learning. When real world images are used to train GAN models, the availability of good datasets is not a problem, there are a large variety of datasets [57,107] that have been widely tested and are well known in the academic community.

Since the first GAN publication [1] GAN architectures have been used for synthesizing real world images. In the original proposed GAN the models were used to generate images replicating MNIST [148], CIFAR-10 [107] and Toronto Face Database (TFD) [149] datasets. The generated images using the original structure were very blurry and did not have good quality. Besides that, the presented results supposed the presentation of the GAN architecture.

One of the first improvements to the original architecture was the DCGAN [69], it proposed structural changes and hyper-parameter tuning respect the first proposed model. The results of the DCGAN showed improvements in the performance and generation of the networks, the generated images were clearer and more recognizable. Despite that, the architecture still suffered from instability and mode collapse.

The WGAN architecture [26] could reduce drastically the mode collapse and instability of the previous models. Thus, later models adapted the loss function of the WGAN along with their respective structural changes in the network.

Recently the ProGAN [3] introduced a new training methodology that achieved an improved performance of the networks. With the new methodology came a huge improvement in the quality of the generated images. The results showed not only a more stable trainings but sharper, with finer details and more diverse images. Due to the particularities of the applied methodology, it can be applied to other architectures, so in later works the ProGAN training methodology will be used as its base.

Following the line of research of ProGAN the StyleGAN [6] was presented. The results produced by the StyleGAN could improve the results of the ProGAN. At this point some generated datasets, e.g. human faces images, were indistinguishable from real images from a human perception. Along with the high quality of the images the StyleGAN proposed style mixing, capable of generating new images combining previous images. This allows to modify image features at a high, medium and low level, allowing the network to disentangle different features of an image, providing more control of the generated images.

One of the main problems of the StyleGAN was the known as *texture sticking*. This caused the generated images to have a certain texture in an absolute position. When interpolating different images it was noticeable that some parts of the images, e.g. the hair of a human face, maintain the same texture in spite of changing its position. The Alias-Free GAN proposed an architecture that suppressed the texture sticking problem. By eliminating the sticking problem, the interpolation of synthesized images is smoothed, generating a continuum of images, not only realistic individually but also as a set. The improvements of the Alias-Free GAN together with the style mixing of StyleGAN allows to create animations of, for example, a human face changing its position, gender or features such as the smile.

The Table 2 summarizes the performance of the presented GAN models during this section. The compared datasets are

Table 2
Performance summary of image generation GANs.

Model	CIFAR-10		CelebA-HQ	FFHQ
	Accuracy ↑	IS ↑	FID ↓	FID ↓
DCGAN	**82.8%**	6.58	–	–
ProGAN	–	**8.80**	7.79	8.04
StyleGAN	–	–	**5.06**	4.40
StyleGAN2	–	–	–	**2.70**
Alias-Free GAN	–	–	–	3.07

MNIST [148], TFD [149], CIFAR-10 [107], CelebA-HQ [3] and Flickr-Faces-HQ (FFHQ) [6]. The used metric for comparing the different variants are accuracy of the models (the higher score the better ↑), IS (the higher score the better ↑) and FID (the lower score the better ↓)

6.2. Image-to-image translation

Taking an image from one domain and converting it to the other domain is known as image-to-image translation. It was first proposed with the Pix2Pix architecture [76], Pix2Pix is based on CGAN following the idea of generating images conditioned on their composition via a label input. With Pix2Pix the networks are capable of learning how the same image is translated between one domain and another. The main drawback that Pix2Pix had was the requirement of having a paired dataset of images in both domains.

Following the steps of Pix2Pix CycleGAN [7], DualGAN [94] and DiscoGAN [101] were developed. These new architectures were based on the cyclic consistency idea. Cyclic consistency was previously used in machine learning [92,93], it is based on the idea that translating an image from one domain to another and then doing the reverse operation will recover the original image. Following this concept the new networks were capable of translating images without a paired dataset. By not needing a paired dataset the number of possible applications of GAN to image-to-image translation increased considerably.

Later on the CSGAN was proposed [135] improving the results of previous architectures. The new proposed loss function achieved better results in image generation, comparing with CycleGAN [7], DualGAN [94], DiscoGAN [101] and PS2MAN [138]. This new architecture results follows the natural progression of the GAN in image-to-image translation and promise an exciting future in what GAN can do.

The image-to-image translation is especially popular in society, because of the applications that have been developed in the last years. With the architecture of the presented GANs the general public is capable, for example, of taking a personal image of themselves and transforming it into one of an old person with his face. This type of applications have become popular in social networks, increasing their visibility even more.

This interaction between society and GAN development is mutually beneficial, the society uses the technological advances of the last years while the academic community gain impact and repercussion. From an academic perspective this interaction should be considered positive and it should be noted that most of the impact of machine learning during the last years have been caused by the publicity given by the mass media and the social networks. Although most of the people are not interested in the technique behind GAN applications they act as a catalyst to make more people interested in artificial intelligence and, ultimately, it will bring more people to academic research in the field.

The Table 3 summarizes the performance of the presented GAN models in image-to-image translation tasks. The data is obtained from [135], where the SSIM (the higher score the better ↑), MSE (the lower score the better ↓), Peak Signal to Noise Ratio

(PSNR) (the higher score the better ↑) and Learned Perceptual Image Patch Similarity (LPIPS) [150,151] (the lower score the better ↓) are computed for different GAN variants. The comparison is made for CUHK [136] and FACADES [137] datasets. The LPIPS is a metric that measure the distance between the real and the generated distribution via perceptual similarity.

6.3. Video generation

GANs have proven to generate state-of-the-art results in image processing. Along with image generation comes the possibility to generate a set of images generating a video. Video generation is a more complex task than image generation. The issues associated with image generation are included in video generation, but the computational cost of training models that can process video is high. In addition, the synthesized videos must be coherent.

One of the particular problems of video is the motion blur generated by the networks [152]. When a video is generated, the tracking of some objects can be difficult, generating fuzziness in some portions of the image. Some works have tried to tackle this problem [153–155], but it is still an open problem.

One of the most popular applications of video generation with GANs is the known as *deep fake*. Deep fake consists in taking a video of a person and changing the face of the human to be someone else. Many works have been developed in the last years in this field [156].

Deep fake is one of the most controversial applications of GAN, the possibility of changing a face in a video allows to generate fake videos that can be used to supplant a person. This problem is magnified in the case of women [157] due to their position in society. Even so, there are some applications of deep fake where it can be beneficial [158], its application still raises doubts in the society. This is why many recent researches have focused on how to detect deep fake videos [159–162].

Other application of GANs to video generation are video-to-video translation, which is indeed the general case of deep fake. Many architectures of this type have been proposed during the last years [163,164].

It should be noted that, in the case of video processing, the standard is to use previous information, such as another video, to generate the synthesized data. Unlike image generation, video generation is more interesting if the new information is conditioned by an external agent. In image processing, the only input was the latent space, but the final images were conditioned by the dataset of the training. When videos are generated, the degree of freedom is extended, enabling the generated data to be less controlled. Controlling the video output is necessary to maintain the coherence of the final output, but it also eases the GAN job, which is significantly more difficult with respect to image processing.

6.4. Image generation from text

Since the introduction of CGAN the capabilities of GANs were expanded. The possibility of constraint the synthesized information that GANs produced made the networks have a wider range

Table 3

Performance summary of image-to-image translation GANs.

Model	CUHK				FACADES			
	SSIM ↑	MSE ↓	PSNR ↑	LPIPS ↓	SSIM ↑	MSE ↓	PSNR ↑	LPIPS ↓
GAN	0.5398	94.8815	28.3628	0.157	0.1378	103.8049	27.9706	0.525
Pix2Pix	0.6056	89.9954	28.5989	0.154	0.2106	**101.9864**	**28.0568**	**0.216**
DualGAN	0.6359	85.5418	28.8351	0.132	0.0324	105.0175	27.9187	0.259
CycleGAN	0.6537	89.6019	28.6351	0.099	0.0678	104.3104	27.9489	0.248
PS2MAN	0.6409	86.7004	28.7779	0.098	0.1764	102.4183	28.032	0.221
CSGAN	**0.6616**	**84.7971**	**28.8693**	**0.094**	**0.2183**	103.7751	27.9715	0.22

of application. By controlling the output of the generations of the networks the applications of them can be much more specific and interesting. One field were GANs have shown to outperform previous techniques in image generation from text [165].

Stacked GANs (StackGAN) [166] was one of the firsts proposed architectures for image generation from text. The architecture splits in two stages, the generation problem, the objective is to divide the main problem in sub-problems that are easier to handle in the network. The known as *Stage-I GAN* is in charge of producing a coarse sketch of the desired image, this way this part of the network focuses on translating the text to a image that fulfills the description. Then, the *Stage-II GAN* takes the generated image from Stage-I GAN, increases its resolution and define the finer details. The StackGAN is able of producing images that match the input description while achieving sharp, high quality samples. Later on the StackGAN++ (StackGAN-v2) [167] was proposed, this new architecture resolved some problems of the original StackGAN, stabilizing its training and improving the overall quality of the synthesized images.

One problem of the StackGAN is that it is highly dependent on the sketch generated by the Stage-I GAN. To solve this Dynamic Memory GAN (DM-GAN) proposed a new technique based on memory networks [168,169] that divides the generation problem in two steps. In the first one a initial image is generated and in the second step a memory network is used to refine the details and produce a high quality image. To connect the memory and the GAN a *response gate* is proposed, by controlling dynamically the flow of information the gate is capable of fusing the information appropriately. The results of the StackGAN shows a higher quality respecting all previous architectures.

Dual Attentional GAN (DualAttn-GAN) proposed a new architecture based on two modules. The Visual Attention Module (VAM) is in charge of taking care of the internal representations of the image information, capturing the global structures and their relationships. The Textual Attention Module (TAM) defines the relations between the text and the image, defining the links between both. Finally a Attention Embedding Module (AEM) fuse the visual with the textual information, concatenating them along with the input features of the image. The results of the DualAttn-GAN shows an improved performance respecting previously used architectures.

Following the general architecture of StackGAN Deep Fusion GAN (DF-GAN) was proposed [170]. The DF-GAN architecture only have one stage of image generation, this backbone synthesized new images conditioned by an input text using only one pair of G and D. Thus being a simpler structure, DF-GAN achieves better performance and efficiency compared with previous variants. The new techniques that DF-GAN proposes are a new fusion module, known as *deep text-image fusion block*, and a new discriminator capable of promoting the generator to synthesize higher quality images without extra networks. The results of the DF-GAN shows an improvement on the quality of the images, without committing to more complex models and improving the efficiency of the previous architectures.

The one-stream information approach followed in DF-GAN was reused in Lightweight Dynamic Conditional GAN (LD-CGAN) [171]. The proposed architecture of the LD-CGAN consists on one G and two independent discriminators. The generator is composed by a Conditional Embedding (CE) that disentangles the features of the input text by using unsupervised learning. Then a Conditional Manipulating Block (CM-B) provides continuously the images features with the compensation information. Finally using the known as Pyramid Attention Refine Block (PAR-B) the generated image is enriched maintaining multiscale context and spatial multiscale features. The results of the architecture not only shows a higher quality image respecting previous methods, but also improves the performance decreasing the number of parameters by 86.8% and the computation time by 94.9%.

The Table 4 summarizes the performance of the presented GAN models during this section. In addition to the mentioned networks the Generative Adversarial Text to Image Synthesis (GAN-INT-CLS) [172] and the Generative Adversarial What-Where Network (GAWWN) [173] are included, both of this networks act as a reference of previous architectures. The compared metrics are HR (the lower score the better ↓), IS (the higher score the better ↑) and FID (the lower score the better ↓). The compared datasets are Common Objects in Context (COCO) [174], Caltech-UCSD Birds (CUB) [175] and Oxford-102 [176].

6.5. Language generation

GANs models have been used during the last years in Natural Language Processing (NLP) tasks. The previously mentioned text-to-image field is one of the applications of GAN where natural language is involved. But there are some applications of GAN completely focused on how to produce new text using the models.

Previous methods to process natural language used the known as *Long Short-Term Memory (LSTM)* [177]. LSTM is capable of maintaining local relationships in space and time, this feature provides the networks the ability of process whole sentences, paragraphs and text while maintaining global coherence. In addition to LSTM the previous methods used Recurrent Neural Network (RNN) to generate new texts [178].

The Text GAN (textGAN) [179] uses LSTM along with Convolutional Neural Network (CNN) to synthesize new text. The proposed method applies the GAN training methodology via the known as *adversarial training*. The textGAN uses a LSTM as the G of the network and a CNN as the D. One of the main problems of the textGAN was the highly entangled features of the network, making the interpolation of different writing styles very difficult.

The textGAN approach to language generation, suffering from the known as *exposure bias*. This bias is caused by the objective function of the network, that focus on maximizing the log likelihood of the prediction. The exposure bias is visible in the inference stage, when the G generates a sequence of words iteratively predicting each word based on the previous ones. The problem comes when the prediction is based on words never seen

Table 4

Performance summary of image generation from text GANs.

Model	COCO			CUB			Oxford-102		
	HR ↓	IS ↑	FID ↓	HR ↓	IS ↑	FID ↓	HR ↓	IS ↑	FID ↓
GAN-INT-CLS	1.89	7.88	–	2.81	2.88	–	1.87	2.66	–
GAWWN	–	–	–	1.99	3.62	–	–	–	–
StackGAN	**1.11**	8.45	–	1.37	3.70	–	**1.13**	3.20	–
StackGAN-v2	1.55	8.30	81.59	**1.19**	4.04	15.30	1.30	3.26	48.68
DM-GAN	–	**30.49**	32.64	–	4.75	16.09	–	–	–
DualAttn-GAN	–	–	–	–	4.59	**14.06**	–	**4.06**	**40.31**
DF-GAN	–	–	**21.42**	–	**5.10**	14.81	–	–	–
LD-CGAN	–	–	–	–	4.18	–	–	3.45	–

before in the training stage. Some works were made to tackle this problem [180] but the Sequence GAN (SeqGAN) [181] is the architecture that betters the results produced.

The G of SeqGAN is trained using a stochastic policy of Reinforcement Learning (RL). The RL reward is calculated by judging a complete sentence made with the G of the model. Then, to compute the intermediate steps a Monte Carlo Search is made [182]. The results of the SeqGAN shows a huge improvement in tasks such as language generation, poem composition and music generation. In addition, the performance of the models shows certain creativity in the synthesized data.

Despite the good results of GAN in NLP tasks during the last years, there have been developed architectures that outperform GANs in language generation. The most successful architecture of this field is the Generative Pre-trained Transformer 3 (GPT-3) [183], which belongs to the GPT-n series. The GPT-3 is a generator model based on the transformer [117] architecture. The extraordinary results presented by the GPT-3 are often very difficult to distinguish from human writing. The emergence of the GPT-3 caused a lower interest in GAN models applied to NLP. Due to the good results of transformers in NLP, the GAN approximation to this field has been losing interest.

6.6. Data augmentation

Other field where GANs have shown to be really useful is in data augmentation. Due to the particularities of the GAN they can be used to obtain more samples of an origin data distribution, replicating its distribution. This way, by using GANs, the number of samples of a dataset can be multiplied.

Traditionally, data augmentation was achieved via transforming the initial data; e.g. cropping, rotating, shearing, or flipping images. One of the main drawbacks of these methods is that they transform the original data by slightly changing their structure, with the usage of GANs for data augmentation the new samples tries to synthesize new data from the original distribution. Instead of changing the samples of the dataset the generated samples of GAN are synthesized from scratch. This way, the new data is replicated by imitating the original data distribution. It should be noted that data augmentation does not necessarily replace other methods of data augmentation, it proposes an alternative that, in many cases, can be used together with other data augmentation algorithms.

For example, the Data Augmentation Optimized for GAN (DAG) [184] proposes an enhanced data augmentation method for GAN, combining it with data transformation such as rotation, flipping or cropping. The DAG shows to improve the performance of data augmentation in GAN models, improving the FID of CGAN, Self-supervised GAN (SSGAN) and CycleGAN. The proposed architecture uses one D for each transformation of the data, but a unique G.

Data augmentation with GANs have been used in cases where obtaining a dataset is difficult. For example, in medical applications there is usually not many information available, in this cases GANs can make the difference. This is why during the last years GANs have been used in medical data augmentation [185–188].

6.7. Other domains

As mentioned before, due to the particularities of the GANs they can be applied to many different fields. One of the main strengths of the machine learning is that it adapts to different situations without substantial changes in its structure. In particular, GAN can be adapted to any type of data distribution as long as there is an available dataset.

6.7.1. GameGAN

One of the most interesting applications of GAN is the presented with the GameGAN [11]. The main purpose of GameGAN is to generate entirely a video game using machine learning. To do so, the complete Model-View-Controller (MVC) software design patterns is replicated using artificial intelligence. The proposed architecture is composed by three different modules.

The *dynamics engine* is in charge of the logic of the whole system, maintaining the global coherence and updating the internal state of the game. The dynamics engine, for example, controls which actions of the game are possible (e.g. eating a fruit in pac-man) and which ones are not (e.g. run through a wall in pac-man). The dynamics engine is composed by an LSTM that updates the state of the game in each frame, the LSTM provides the network way to control the previous states of the game to calculate the new information of the subsequent frames. This way, the network can access to the complete history of the game, maintaining the consistency of the system.

To save the state of the game a *memory module* is used. This module focus on maintain long-term consistency of the game scene. When the game is being played there are different elements of the scene that not always are visible, with the memory module these elements are consistent over the time. This memory remembers the generated static elements of the game. The memory module is implemented by using Neural Turing Machine (NTM) [189].

The third module that composes the system is the *rendering engine*, it is in charge of generating a visualization of the current state of the game. This module focuses on representing the different elements of the game realistically, producing disentangled scenes. The rendering engine is composed by transposed convolution layers that are initially trained using an autoencoder architecture to warm up the system and then they train along with the rest of the modules.

The adversarial training of GameGAN has three types of discriminators. The *single image discriminator* evaluates the quality of each generated frame, judging how realistic it is. The *action-conditioned discriminator* determines if two consecutive frames

are consistent with respect the input of the player. Finally the *temporal discriminator* maintains the long-term consistency of the scene, preventing elements from appearing or disappearing randomly.

One of the basis of GameGAN is the disentangling of dynamic and static elements of the game. The static elements of a game could be, for example, walls while the dynamics elements of a game are elements such as nonplayable characters. By disentangling both types of elements, the game behavior is more interpretable for the model.

Finally, GameGAN introduces a warm-up phase where certain real frames are introduced in the network during the first epoch of the training. Then the frequency of real frames is reduced little by little until it disappears. This way the first epochs of the training, that are usually the most complex in the network, are controlled and progressively the GAN gains more control over the output. This helps the network to understand the problem.

6.7.2. Medical imaging GANs

One of the most popular application of the GAN architecture is to enlarge datasets. The objective of synthesizing new data is to produce larger datasets that improve the performance of machine learning models, which are very sensible with the number of samples used in their training.

There are many fields where data augmentation can be applied, but in medical imaging to augment data have certain benefits due to the particularities of the problem. First, the medical datasets are usually small, because of the cost of obtaining the images, most of the time it is necessary to use measurement and recording machines such as radiography, magnetic resonance or ultrasound. But, in addition to the cost of obtaining these images there also exists ethical and legal problems related to the nature of the data. Most of the time, obtaining images that expose the health status of different people is impossible, which leads to even more lack of available data.

It should be noted that one of the benefits of generating data with GAN is that the new samples do not belong to any real person.

Because of all these factors, there has been lots of GAN works related to the medical imaging field [190–195]. In addition, the work of Chen et al. [196] analyzes the evolution of the field of medical data augmentation and suggests that the research in this field remains strong in the year 2021, despite that the fact that from 2019 onwards the number of published works have been the same.

6.7.3. GANs in agriculture

Similar to the medical imaging field, obtaining images to train the computer vision models of agricultural image analysis is not an easy task. These models benefit from having large-scale balanced datasets but the cost of obtaining high quality labeled data makes the data augmentation a crucial task in these datasets.

Many different GAN models have been applied to agricultural data, such as [197–199]. These works aim to generate new images of plant with different diseases, augmenting the number of samples by using GAN.

In these cases the use of GAN improves the results of the machine learning models by enlarging the number of available data. The agricultural images have different particularities that make the analysis of them a difficult task. For example the biological variability between two samples of the same species makes crucial to have many different samples to learn all the modes of the data. In particular, the same leaf of a fruit can drastically differ from one individual to another.

Other important factor is that the labeling of the data can be very costly, specially for specific applications such as the disease detection of certain plant, e.g. tomato leaf [197].

In addition, the environment where the images are taken, most of the time in crops, can lead to many variance in the images, such as lighting changes or object occlusion.

6.7.4. Drug discovery using GANs

The process of discovering and designing new drugs has recently been impulsed by the field of Deep Learning [200,201]. In particular, GANs are an useful technique to synthesize new useful samples of data. In the drug environment, the GAN architecture can process the drug compound using graphs or Simplified Molecular Input Line Entry Specification (SMILES), to then generate synthetic samples of drugs.

Due to the flexibility that ANNs have in terms of operating with different data types, it is possible to use the same architectures in different fields. In this case the overall GAN design can be adapted to molecular data, being able to transfer the same principles of the image generation to new data types.

The research followed by Kadurin et al. [202,203] generates new drug compounds for anticancer therapy, using biological and chemical datasets. In particular, in [203] it is used an Adversarial Autoencoder that uses molecular fingerprints as inputs of the network. By using this architecture the researches are able of define the desired properties of the synthesized drugs. Some of the new synthetic drugs discovered by the Deep Learning architecture corresponded with previous known anticancer drugs. This led the researches to suggest that the remaining unknown drugs generated by the GAN could be used to further study their properties.

The work presented in [204] proposes the generation of new drugs combining GANs with reinforcement learning techniques. In particular, the proposed G takes as input a random latent space and process it with RNN to produce a sequence of drug by using SMILES representation. The D on its side uses a 1 dimensional CNN to distinguish the real data from the synthesized one. The results of the paper suggest that the new drugs discovered were unique and diverse. This may alleviate the first phases of drug development, which are very expensive in terms of time.

The Federated Generative Adversarial Network for Graph-based Molecule Drug Discovery (FL-DISCO) architecture [205] aims to combine the generation potential of GAN with the processing of molecules using graphs of the Graph Neural Networks while maintaining the privacy of the data using Federated Learning [206]. By using graph representation of the molecules instead of SMILES as previous works, the represented samples have more realistic structures, maintaining structural relationships of the connected atoms of the molecules. The Federated Learning framework is based on using different clients to train a specific neural network model, each client has its respective portion of the data, which uses to train the network. This way each client knows a portion of the data and uses it to update the central model, but it maintains the privacy due to the fact that the clients are not able to communicate with each other. The results of this research show progress in terms of novelty and diversity of the synthesized drugs respect previous works.

7. Discussion

Since their introduction in 2014 GANs have been the most important generative architecture in computer vision. The results provided by the developed GANs were notoriously better than previous architectures, such as Variational Autoencoders. This leaded to a constant improvement of the model, solving problems like stabilization or mode collapse.

With the introduction of the Diffusion models [207–209], the results of GANs have been surpassed by this new models solving some of its most important problems. Some aspects in which

diffusion models outperform GANs are better stability, they do not suffer from mode collapse and they provide more diverse results. This is mainly caused because of the fact that they are likelihood-based [210]. Despite the better results of diffusion models they still have shortcomings in some aspects such as the cost of synthesizing new samples, which makes them difficult to being applied in real-time problems.

In [211] it was developed a diffusion model to perform an image-to-image translation. The results showed in this research show that their solution outperforms GANs without special attention to the hyper-parameter tuning or any kind of sophisticated technique or loss function. Moreover this research shows the great stability of the diffusion model architecture.

Despite the fact that Diffusion models are a novel architecture with not many works published, it is a very potential architecture to surpass GAN results in a near future. At present, there are not enough results or applications of diffusion models to data generation, but the potential of this new architecture could lead to a significant improvement in the results of data synthesis. We consider that this models could replace GANs because of their stability and not needing fine-tuning in their hyperparameters.

Other new architectures have been used to enhance the results of GANs, such as transformers, to improve their results. Transformer architecture is a time-series-based architecture that adopts the self-attention layers [117] making possible to design larger models. Transformers have been used as the base neural model of the G and D of the GAN architecture, improving the performance of the model.

The TransGAN [212] presents a GAN architecture free of convolutions that makes possible to generate high resolution images by using transformer in both G and D of the GAN. The results of the article shows an improved results respect to the IS and FID on CIFAR-10 dataset [107].

Another work that showcases the interaction between GANs and transformers is the one presented in [213]. This work uses the generative model to predict pedestrian paths, using the memory that the transformer architecture has. In this sense, the GAN makes possible to train the network to predict future paths of pedestrians, while the transformer provides the memory to process an historical sequence of the latest movements.

8. Conclusion

This report summarizes the recent progress of GANs, going from the basic principles in which GAN are sustained to the most innovative architectures of the last years. In addition, the different problems that GANs can suffer are categorized and the most common evaluation metrics are explained and discussed.

Respect the recent progress in the field, a taxonomy for the GAN variants is proposed. The researches are divided in two groups, one with the GANs that focus in architecture optimization and the other with the GANs that focus in objective function optimization. Despite being two separate groups of variants, it should be noted that the different researches benefit from the progress of the rest. These ecosystem where there are various approaches for GAN development is connected with the main problems that are reviewed in this survey, since normally each research focus in trying to solve a certain problematic of previous researches.

Finally the different application of the GANs during the last years are summarized. The different applications of GAN are influenced by the development of the field, its impact in the society and in the industry. We conclude with a comparison between the different architectures performance to provide a quantitative view of the evolution of GANs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014.

[2] J. Cheng, Y. Yang, X. Tang, N. Xiong, Y. Zhang, F. Lei, Generative adversarial networks: A literature review., KSII Trans. Internet Inf. Syst. 14 (12) (2020).

[3] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of GANs for improved quality, stability, and variation, 2018.

[4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein GANs, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS '17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 5769–5779.

[5] J. Xu, X. Ren, J. Lin, X. Sun, Diversity-promoting GAN: A cross-entropy based generative adversarial network for diversified text generation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 3940–3949.

[6] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, 2019.

[7] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2242–2251.

[8] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, 2018.

[9] M. Zhu, P. Pan, W. Chen, Y. Yang, DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019.

[10] Y. Li, M. Min, D. Shen, D. Carlson, L. Carin, Video generation from text, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018, p. 1.

[11] S.W. Kim, Y. Zhou, J. Philion, A. Torralba, S. Fidler, Learning to simulate dynamic environments with gamegan, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1231–1240.

[12] D.H. Ackley, G.E. Hinton, T.J. Sejnowski, A learning algorithm for Boltzmann machines, Cogn. Sci. 9 (1) (1985) 147–169.

[13] D. Bank, N. Koenigstein, R. Giryes, Autoencoders, 2021.

[14] A. van den Oord, N. Kalchbrenner, Pixel RNN, in: ICML, 2016.

[15] Y. Sun, L. Xu, L. Guo, Y. Li, Y. Wang, A comparison study of VAE and GAN for software fault prediction, in: S. Wen, A. Zomaya, L.T. Yang (Eds.), Algorithms and Architectures for Parallel Processing, Springer International Publishing, Cham, 2020, pp. 82–96.

[16] M. Wiatrak, S.V. Albrecht, Stabilizing generative adversarial network training: A survey, 2019, arXiv.

[17] H. Thanh-Tung, T. Tran, S. Venkatesh, Improving generalization and stability of generative adversarial networks, 2019.

[18] X. Mao, Q. Li, H. Xie, R.Y. Lau, Z. Wang, S. Paul Smolley, Least squares generative adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2017.

[19] Bhagyashree, V. Kushwaha, G.C. Nandi, Study of prevention of mode collapse in generative adversarial network (GAN), in: 2020 IEEE 4th Conference on Information Communication Technology, CICT, 2020, pp. 1–6.

[20] D. Bang, H. Shim, MGGAN: Solving mode collapse using manifold guided training, 2018.

[21] S. Adiga, M.A. Attia, W.-T. Chang, R. Tandon, On the tradeoff between mode collapse and sample quality in generative adversarial networks, in: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2018, pp. 1184–1188.

[22] D. Bau, J.-Y. Zhu, J. Wulff, W. Peebles, H. Strobelt, B. Zhou, A. Torralba, Seeing what a GAN cannot generate, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2019.

[23] R. Durall, A. Chatzimichailidis, P. Labus, J. Keuper, Combating mode collapse in GAN training: An empirical analysis using hessian eigenvalues, 2020.

[24] H. Thanh-Tung, T. Tran, Catastrophic forgetting and mode collapse in GANs, in: 2020 International Joint Conference on Neural Networks, IJCNN, 2020, pp. 1–10.

[25] A. Aggarwal, M. Mittal, G. Battineni, Generative adversarial network: An overview of theory and applications, Int. J. Inf. Manage. Data Insights 1 (1) (2021) 100004.

[26] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, 2017.

[27] B. Ghosh, I.K. Dutta, M. Totaro, M. Bayoumi, A survey on the progression and performance of generative adversarial networks, in: 2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT, 2020, pp. 1–8.

[28] Z. Wang, Q. She, T.E. Ward, Generative adversarial networks in computer vision: A survey and taxonomy, 2020.

[29] H. Alqahtani, M. Kavakli-Thorne, D.G. Kumar Ahuja, Applications of generative adversarial networks (GANs): An updated review, Arch. Comput. Methods Eng. 28 (2019).

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

[31] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, F.-Y. Wang, Generative adversarial networks: introduction and outlook, IEEE/CAA J. Autom. Sin. 4 (4) (2017) 588–598.

[32] V. Sampath, I. Maurtua, J.J.A. Martín, A. Gutierrez, A survey on generative adversarial networks for imbalance problems in computer vision tasks, J. Big Data 8 (1) (2021) 1–59.

[33] X. Wu, K. Xu, P. Hall, A survey of image synthesis and editing with generative adversarial networks, Tsinghua Sci. Technol. 22 (6) (2017) 660–674.

[34] Z. Pan, W. Yu, B. Wang, H. Xie, V.S. Sheng, J. Lei, S. Kwong, Loss functions of generative adversarial networks (GANs): opportunities and challenges, IEEE Trans. Emerg. Top. Comput. Intell. 4 (4) (2020) 500–522.

[35] J. Gui, Z. Sun, Y. Wen, D. Tao, J. Ye, A review on generative adversarial networks: Algorithms, theory, and applications, 2020.

[36] H. Zhang, Z. Le, Z. Shao, H. Xu, J. Ma, MFF-GAN: An unsupervised generative adversarial network with adaptive and gradient joint constraints for multi-focus image fusion, Inf. Fusion 66 (2021) 40–53.

[37] R. Liu, Y. Ge, C.L. Choi, X. Wang, H. Li, DivCo: Diverse conditional image synthesis via contrastive generative adversarial network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 16377–16386.

[38] D.M. De Silva, G. Poravi, A review on generative adversarial networks, in: 2021 6th International Conference for Convergence in Technology (I2CT), 2021, pp. 1–4.

[39] L. Metz, B. Poole, D. Pfau, J. Sohl-Dickstein, Unrolled generative adversarial networks, 2017.

[40] S. Suh, H. Lee, P. Lukowicz, Y.O. Lee, CEGAN: Classification enhancement generative adversarial networks for unraveling data imbalance problems, Neural Netw. 133 (2021) 69–86.

[41] J. Nash, Non-cooperative games, Ann. of Math. (1951) 286–295.

[42] F. Farnia, A. Ozdaglar, GANs may have no Nash equilibria, 2020.

[43] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, Adv. Neural Inf. Process. Syst. 30 (2017).

[44] Á. González-Prieto, A. Mozo, E. Talavera, S. Gómez-Canaval, Dynamics of Fourier modes in torus generative adversarial networks, Mathematics 9 (4) (2021).

[45] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANs, 2016.

[46] Z. Zhang, C. Luo, J. Yu, Towards the gradient vanishing, divergence mismatching and mode collapse of generative adversarial nets, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 2377–2380.

[47] H.D. Meulemeester, J. Schreurs, M. Fanuel, B.D. Moor, J.A.K. Suykens, The bures metric for generative adversarial networks, 2021.

[48] W. Li, L. Fan, Z. Wang, C. Ma, X. Cui, Tackling mode collapse in multi-generator GANs with orthogonal vectors, Pattern Recognit. 110 (2021) 107646.

[49] I. Goodfellow, NIPS 2016 tutorial: Generative adversarial networks, 2017.

[50] S. Pei, R.Y. Da Xu, G. Meng, dp-GAN: Alleviating mode collapse in GAN via diversity penalty module, 2021, arXiv preprint arXiv:2108.02353.

[51] J. Su, GAN-QP: A novel GAN framework without gradient vanishing and Lipschitz constraint, 2018.

[52] Y. Zuo, G. Avraham, T. Drummond, Improved training of generative adversarial networks using decision forests, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, 2021, pp. 3492–3501.

[53] S. Liu, O. Bousquet, K. Chaudhuri, Approximation and convergence properties of generative adversarial learning, 2017.

[54] S.A. Barnett, Convergence problems with generative adversarial networks (GANs), 2018.

[55] A. Borji, Pros and cons of GAN evaluation measures, Comput. Vis. Image Underst. 179 (2019) 41–65.

[56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, 2015.

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.

[58] S. Nowozin, B. Cseke, R. Tomioka, f-GAN: Training generative neural samplers using variational divergence minimization, 2016.

[59] S. Gurumurthy, R.K. Sarvadevabhatla, V.B. Radhakrishnan, DeLiGAN: Generative adversarial networks for diverse and limited data, 2017.

[60] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, T. Aila, Alias-free generative adversarial networks, 2021, arXiv preprint arXiv:2106.12423.

[61] G. Daras, A. Odena, H. Zhang, A.G. Dimakis, Your local GAN: Designing two dimensional local attention mechanisms for generative models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14531–14539.

[62] Z. Wang, E. Simoncelli, A. Bovik, Multiscale structural similarity for image quality assessment, in: The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003, Vol. 2, 2003, pp. 1398–1402, Vol.2.

[63] K. Kurach, M. Lucic, X. Zhai, M. Michalski, S. Gelly, The GAN landscape: Losses, architectures, regularization, and normalization, 2019.

[64] E.L. Lehmann, J.P. Romano, Testing Statistical Hypotheses, Springer Science & Business Media, 2006.

[65] D. Lopez-Paz, M. Oquab, Revisiting classifier two-sample tests, 2018.

[66] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.

[67] W. Bounliphone, E. Belilovsky, M.B. Blaschko, I. Antonoglou, A. Gretton, A test of relative similarity for model selection in generative models, 2016.

[68] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, B. Póczos, MMD GAN: Towards deeper understanding of moment matching network, 2017.

[69] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[70] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, K. Tunyasuvunakool, O. Ronneberger, R. Bates, A. Žídek, A. Bridgland, et al., High accuracy protein structure prediction using deep learning, in: Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book), Vol. 22, 2020, p. 24.

[71] J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, 2015.

[72] R. Ayachi, M. Afif, Y. Said, M. Atri, Strided convolution instead of max pooling for memory efficiency of convolutional neural networks, in: M.S. Bouhlel, S. Rovetta (Eds.), Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT'18), Vol. 1, Springer International Publishing, Cham, 2020, pp. 234–243.

[73] Y. Li, N. Xiao, W. Ouyang, Improved boundary equilibrium generative adversarial networks, IEEE Access 6 (2018) 11342–11348.

[74] S. Wu, G. Li, L. Deng, L. Liu, D. Wu, Y. Xie, L. Shi, L1 norm batch normalization for efficient training of deep neural networks, IEEE Trans. Neural Netw. Learn. Syst. 30 (7) (2019) 2043–2051.

[75] D.H. Hubel, T.N. Wiesel, Receptive fields of single neurones in the cat's striate cortex, J. Physiol. 148 (3) (1959) 574–591.

[76] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014.

[77] M. Loey, G. Manogaran, N.E.M. Khalifa, A deep transfer learning model with classical data augmentation and cgan to detect covid-19 from chest ct radiography digital images, Neural Comput. Appl. (2020) 1–13.

[78] Y. Ma, X. Chen, W. Zhu, X. Cheng, D. Xiang, F. Shi, Speckle noise reduction in optical coherence tomography images based on edge-sensitive cGAN, Biomed. Opt. Express 9 (11) (2018) 5129–5146.

[79] Y. Li, R. Fu, X. Meng, W. Jin, F. Shao, A SAR-to-optical image translation method based on conditional generation adversarial network (cGAN), IEEE Access 8 (2020) 60338–60343.

[80] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, Infogan: Interpretable representation learning by information maximizing generative adversarial nets, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 2180–2188.

[81] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier gans, in: International Conference on Machine Learning, PMLR, 2017, pp. 2642–2651.

[82] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (3) (1948) 379–423.

[83] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[84] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[85] Y. Zhou, T.L. Berg, Learning temporal transformations from time-lapse videos, in: European Conference on Computer Vision, Springer, 2016, pp. 262–277.

[86] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: European Conference on Computer Vision, Springer, 2016, pp. 694–711.

[87] M. Liu, J. Zhu, A. Tao, J. Kautz, B. Catanzaro, High-resolution image synthesis and semantic manipulation with conditional gans, in: ICCV, 2017.

[88] Y. Qu, Y. Chen, J. Huang, Y. Xie, Enhanced pix2pix dehazing network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8160–8168.

[89] M. Mori, T. Fujioka, L. Katsuta, Y. Kikuchi, G. Oda, T. Nakagawa, Y. Kitazume, K. Kubota, U. Tateishi, Feasibility of new fat suppression for breast MRI using pix2pix, Jpn. J. Radiol. 38 (11) (2020) 1075–1081.

[90] W. Pan, C. Torres-Verdín, M.J. Pyrcz, Stochastic pix2pix: a new machine learning method for geophysical and well conditioning of rule-based channel reservoir models, Natural Resour. Res. 30 (2) (2021) 1319–1345.

[91] M. Drob, RF PIX2PIX unsupervised wi-fi to video translation, 2021, arXiv preprint arXiv:2102.09345.

[92] N. Sundaram, T. Brox, K. Keutzer, Dense point trajectories by gpu-accelerated large displacement optical flow, in: European Conference on Computer Vision, Springer, 2010, pp. 438–451.

[93] Z. Kalal, K. Mikolajczyk, J. Matas, Forward-backward error: Automatic detection of tracking failures, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 2756–2759.

[94] Z. Yi, H. Zhang, P. Tan, M. Gong, Dualgan: Unsupervised dual learning for image-to-image translation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2849–2857.

[95] J. Ye, Y. Ji, X. Wang, X. Gao, M. Song, Data-free knowledge amalgamation via group-stack dual-gan, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12516–12525.

[96] D. Prokopenko, J.V. Stadelmann, H. Schulz, S. Renisch, D.V. Dylov, Synthetic CT generation from MRI using improved DualGAN, 2019, arXiv preprint arXiv:1909.08942.

[97] W. Liang, D. Ding, G. Wei, An improved DualGAN for near-infrared image colorization, Infrared Phys. Technol. 116 (2021) 103764.

[98] C.L.M. Veillon, N. Obin, A. Roebel, Towards end-to-end F0 voice conversion based on dual-GAN with convolutional wavelet kernels, 2021, arXiv preprint arXiv:2104.07283.

[99] F. Yger, A. Rakotomamonjy, Wavelet kernel learning, Pattern Recognit. 44 (10–11) (2011) 2614–2629.

[100] Z. Luo, J. Chen, T. Takiguchi, Y. Ariki, Emotional voice conversion using dual supervised adversarial networks with continuous wavelet transform f0 features, IEEE/ACM Trans. Audio Speech Lang. Process. 27 (10) (2019) 1535–1548.

[101] T. Kim, M. Cha, H. Kim, J.K. Lee, J. Kim, Learning to discover cross-domain relations with generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1857–1865.

[102] C.R.A. Chaitanya, A.S. Kaplanyan, C. Schied, M. Salvi, A. Lefohn, D. Nowrouzezahrai, T. Aila, Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder, ACM Trans. Graph. 36 (4) (2017) 1–12.

[103] I.A. Luchnikov, A. Ryzhov, P.-J. Stas, S.N. Filippov, H. Ouerdane, Variational autoencoder reconstruction of complex many-body physics, Entropy 21 (11) (2019) 1091.

[104] J. Mehta, A. Majumdar, Rodeo: robust de-aliasing autoencoder for real-time medical image reconstruction, Pattern Recognit. 63 (2017) 499–510.

[105] S. Hicsonmez, N. Samet, E. Akbas, P. Duygulu, GANILLA: Generative adversarial networks for image to illustration translation, Image Vis. Comput. 95 (2020) 103886.

[106] A.A. Rusu, N.C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks, 2016, arXiv preprint arXiv:1606.04671.

[107] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, 2009.

[108] H. Yang, J. Liu, L. Zhang, Y. Li, H. Zhang, ProEGAN-MS: A progressive growing generative adversarial networks for electrocardiogram generation, IEEE Access 9 (2021) 52089–52100.

[109] V. Bhagat, S. Bhaumik, Data augmentation using generative adversarial networks for pneumonia classification in chest xrays, in: 2019 Fifth International Conference on Image Information Processing, ICIIP, IEEE, 2019, pp. 574–579.

[110] L. Liu, Y. Zhang, J. Deng, S. Soatto, Dynamically grown generative adversarial networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 8680–8687.

[111] T. Sainburg, M. Thielk, B. Theilman, B. Migliori, T. Gentner, Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions, 2018, arXiv preprint arXiv:1807.06650.

[112] S. Laine, Feature-Based Metrics for Exploring the Latent Space of Generative Models, ICLR Workshop Poster, 2018.

[113] X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.

[114] M. Tancik, P.P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J.T. Barron, R. Ng, Fourier features let networks learn high frequency functions in low dimensional domains, 2020, arXiv preprint arXiv:2006.10739.

[115] R. Xu, X. Wang, K. Chen, B. Zhou, C.C. Loy, Positional encoding as spatial inductive bias in gans, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13569–13578.

[116] H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, Self-attention generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 7354–7363.

[117] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[118] A. Brock, J. Donahue, K. Simonyan, Large scale GAN training for high fidelity natural image synthesis, 2018, arXiv preprint arXiv:1809.11096.

[119] A.G. Dimakis, P.B. Godfrey, Y. Wu, M.J. Wainwright, K. Ramchandran, Network coding for distributed storage systems, IEEE Trans. Inform. Theory 56 (9) (2010) 4539–4551.

[120] Y. Chen, G. Li, C. Jin, S. Liu, T. Li, SSD-GAN: Measuring the realness in the spatial and spectral domains, 2020, arXiv preprint arXiv:2012.05535.

[121] P. Benioff, The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by turing machines, J. Stat. Phys. 22 (5) (1980) 563–591.

[122] E.R. MacQuarrie, C. Simon, S. Simmons, E. Maine, The emerging commercial landscape of quantum computing, Nat. Rev. Phys. 2 (11) (2020) 596–598.

[123] Y. Cao, J. Romero, J.P. Olson, M. Degroote, P.D. Johnson, M. Kieferová, I.D. Kivlichan, T. Menke, B. Peropadre, N.P. Sawaya, et al., Quantum chemistry in the age of quantum computing, Chem. Rev. 119 (19) (2019) 10856–10915.

[124] S.A. Stein, B. Baheri, R.M. Tischio, Y. Mao, Q. Guan, A. Li, B. Fang, S. Xu, Qugan: A generative adversarial network through quantum states, 2020, arXiv preprint arXiv:2010.09036.

[125] M.Y. Niu, A. Zlokapa, M. Broughton, S. Boixo, M. Mohseni, V. Smelyanskyi, H. Neven, Entangling quantum generative adversarial networks, 2021, arXiv preprint arXiv:2105.00080.

[126] W.W. Ng, J. Hu, D.S. Yeung, S. Yin, F. Roli, Diversified sensitivity-based undersampling for imbalance classification problems, IEEE Trans. Cybern. 45 (11) (2014) 2402–2412.

[127] E. Ramentol, Y. Caballero, R. Bello, F. Herrera, SMOTE-RS B*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory, Knowl. Inf. Syst. 33 (2) (2012) 245–265.

[128] Z. Pan, F. Yuan, J. Lei, W. Li, N. Ling, S. Kwong, MIEGAN: Mobile image enhancement via a multi-module cascade neural network, IEEE Trans. Multimed. 24 (2021) 519–533.

[129] G. Qi, Loss-sensitive generative adversarial networks on lipschitz densities, 2017, CoRR abs/1701.06264. arXiv preprint arXiv:1701.06264.

[130] L. Weng, From gan to wgan, 2019, arXiv preprint arXiv:1904.08994.

[131] J. Cao, L. Mo, Y. Zhang, K. Jia, C. Shen, M. Tan, Multi-marginal wasserstein gan, Adv. Neural Inf. Process. Syst. 32 (2019) 1776–1786.

[132] Y. Xiangli, Y. Deng, B. Dai, C.C. Loy, D. Lin, Real or not real, that is the question, 2020, arXiv preprint arXiv:2002.05512.

[133] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, 2018, arXiv preprint arXiv:1802.05957.

[134] T. Salimans, D.P. Kingma, Weight normalization: A simple reparameterization to accelerate training of deep neural networks, Adv. Neural Inf. Process. Syst. 29 (2016) 901–909.

[135] K.B. Kancharagunta, S.R. Dubey, Csgan: Cyclic-synthesized generative adversarial network for image-to-image transformation, 2019, arXiv preprint arXiv:1901.03554.

[136] X. Wang, X. Tang, Face photo-sketch synthesis and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2008) 1955–1967.

[137] R. Tyleček, R. Šára, Spatial pattern templates for recognition of objects with regular structure, in: German Conference on Pattern Recognition, Springer, 2013, pp. 364–374.

[138] L. Wang, V. Sindagi, V. Patel, High-quality facial photo-sketch synthesis using multi-adversarial networks, in: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), IEEE, 2018, pp. 83–90.

[139] N. Barzilay, T.B. Shalev, R. Giryes, MISS GAN: A multi-IlluStrator style generative adversarial network for image to illustration translation, Pattern Recognit. Lett. (2021).

[140] S.W. Park, J. Kwon, Sphere generative adversarial network based on geometric moment matching, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4292–4301.

[141] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4681–4690.

[142] H. Zhang, T. Zhu, X. Chen, L. Zhu, D. Jin, P. Fei, Super-resolution generative adversarial network (SRGAN) enabled on-chip contact microscopy, J. Phys. D: Appl. Phys. 54 (39) (2021) 394005.

[143] O. Dehzangi, S.H. Gheshlaghi, A. Amireskandari, N.M. Nasrabadi, A. Rezai, OCT image segmentation using neural architecture search and SRGAN, in: 2020 25th International Conference on Pattern Recognition, ICPR, IEEE, 2021, pp. 6425–6430.

[144] S. Zhao, Y. Fang, L. Qiu, Deep learning-based channel estimation with SRGAN in OFDM systems, in: 2021 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2021, pp. 1–6.

[145] B. Liu, J. Chen, A super resolution algorithm based on attention mechanism and SRGAN network, IEEE Access (2021).

[146] A. Genevay, G. Peyré, M. Cuturi, GAN and VAE from an optimal transport point of view, 2017, arXiv preprint arXiv:1706.01807.

[147] E. Denton, A. Hanna, R. Amironesei, A. Smart, H. Nicole, M.K. Scheuerman, Bringing the people back in: Contesting benchmark machine learning datasets, 2020, arXiv preprint arXiv:2007.07399.

[148] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[149] J. Susskind, A. Anderson, G.E. Hinton, The Toronto Face Dataset, Tech. Rep., Technical Report UTML TR 2010-001, U. Toronto, 2010.

[150] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595.

[151] J. Lin, Y. Xia, T. Qin, Z. Chen, T.-Y. Liu, Conditional image-to-image translation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5524–5532.

[152] Q. Guo, W. Feng, R. Gao, Y. Liu, S. Wang, Exploring the effects of blur and deblurring to visual object tracking, IEEE Trans. Image Process. 30 (2021) 1812–1824.

[153] K. Zhang, W. Luo, Y. Zhong, L. Ma, B. Stenger, W. Liu, H. Li, Deblurring by realistic blurring, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2737–2746.

[154] M.A. Younus, T.M. Hasan, Effective and fast deepfake detection method based on haar wavelet transform, in: 2020 International Conference on Computer Science and Software Engineering, CSASE, IEEE, 2020, pp. 186–190.

[155] X. Ren, Z. Qian, Q. Chen, Video deblurring by fitting to test data, 2020, arXiv preprint arXiv:2012.05228.

[156] M. Westerlund, The emergence of deepfake technology: A review, Technol. Innov. Manage. Rev. 9 (11) (2019).

[157] V.C. Martínez, G.P. Castillo, Historia del "fake" audiovisual: "deepfake" y la mujer en un imaginario falsificado y perverso, Hist. Comun. Soc. 24 (2) (2019) 55.

[158] A.O. Kwok, S.G. Koh, Deepfake: A social construction of technology perspective, Curr. Issues Tour. 24 (13) (2021) 1798–1802.

[159] P. Korshunov, S. Marcel, Vulnerability assessment and detection of deepfake videos, in: 2019 International Conference on Biometrics, ICB, IEEE, 2019, pp. 1–6.

[160] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, C. Canton Ferrer, The deepfake detection challenge dataset, 2020, arXiv e-prints arXiv–2006.

[161] N. Carlini, H. Farid, Evading deepfake-image detectors with white- and black-box attacks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 658–659.

[162] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, N. Yu, Multi-attentional deepfake detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2185–2194.

[163] Y. Chen, Y. Pan, T. Yao, X. Tian, T. Mei, Mocycle-gan: Unpaired video-to-video translation, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 647–655.

[164] A. Bansal, S. Ma, D. Ramanan, Y. Sheikh, Recycle-gan: Unsupervised video retargeting, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 119–135.

[165] L. Kurup, M. Narvekar, R. Sarvaiya, A. Shah, Evolution of neural text generation: Comparative analysis, in: Advances in Computer, Communication and Computational Sciences, Springer, 2021, pp. 795–804.

[166] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D.N. Metaxas, Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5907–5915.

[167] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D.N. Metaxas, Stackgan++: Realistic image synthesis with stacked generative adversarial networks, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2018) 1947–1962.

[168] C. Gulcehre, S. Chandar, K. Cho, Y. Bengio, Dynamic neural turing machine with soft and hard addressing schemes, 2016, arXiv preprint arXiv:1607.00036.

[169] J. Weston, S. Chopra, A. Bordes, Memory networks, 2014, arXiv preprint arXiv:1410.3916.

[170] M. Tao, H. Tang, S. Wu, N. Sebe, X.-Y. Jing, F. Wu, B. Bao, Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis, 2020, arXiv preprint arXiv:2008.05865.

[171] L. Gao, D. Chen, Z. Zhao, J. Shao, H.T. Shen, Lightweight dynamic conditional GAN with pyramid attention for text-to-image synthesis, Pattern Recognit. 110 (2021) 107384.

[172] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, Generative adversarial text to image synthesis, in: International Conference on Machine Learning, PMLR, 2016, pp. 1060–1069.

[173] S.E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, H. Lee, Learning what and where to draw, Adv. Neural Inf. Process. Syst. 29 (2016) 217–225.

[174] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[175] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200–2011 dataset, 2011.

[176] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, IEEE, 2008, pp. 722–729.

[177] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[178] A.M. Dai, Q.V. Le, Semi-supervised sequence learning, Adv. Neural Inf. Process. Syst. 28 (2015) 3079–3087.

[179] Y. Zhang, Z. Gan, L. Carin, Generating text via adversarial training, in: NIPS Workshop on Adversarial Training, Vol. 21, academia. edu, 2016, pp. 21–32.

[180] S. Bengio, O. Vinyals, N. Jaitly, N. Shazeer, Scheduled sampling for sequence prediction with recurrent neural networks, 2015, arXiv preprint arXiv:1506.03099.

[181] L. Yu, W. Zhang, J. Wang, Y. Yu, Seqgan: Sequence generative adversarial nets with policy gradient, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017.

[182] C.B. Browne, E. Powley, D. Whitehouse, S.M. Lucas, P.I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, S. Colton, A survey of monte carlo tree search methods, IEEE Trans. Comput. Intell. AI Games 4 (1) (2012) 1–43.

[183] L. Floridi, M. Chiriatti, GPT-3: Its nature, scope, limits, and consequences, Minds Mach. 30 (4) (2020) 681–694.

[184] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, T.-K. Nguyen, N.-M. Cheung, On data augmentation for GAN training, IEEE Trans. Image Process. 30 (2021) 1882–1897.

[185] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, Synthetic data augmentation using GAN for improved liver lesion classification, in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), IEEE, 2018, pp. 289–293.

[186] D. Kiyasseh, G.A. Tadesse, L. Thwaites, T. Zhu, D. Clifton, et al., Plethaugment: Gan-based ppg augmentation for medical diagnosis in low-resource settings, IEEE J. Biomed. Health Inf. 24 (11) (2020) 3226–3235.

[187] C. Qi, J. Chen, G. Xu, Z. Xu, T. Lukasiewicz, Y. Liu, SAG-GAN: Semi-supervised attention-guided GANs for data augmentation on medical images, 2020, arXiv preprint arXiv:2011.07534.

[188] M. Hammami, D. Friboulet, R. Kechichian, Cycle GAN-based data augmentation for multi-organ detection in CT images via yolo, in: 2020 IEEE International Conference on Image Processing, ICIP, IEEE, 2020, pp. 390–393.

[189] A. Graves, G. Wayne, I. Danihelka, Neural turing machines, 2014, arXiv preprint arXiv:1410.5401.

[190] P. Guo, P. Wang, J. Zhou, V.M. Patel, S. Jiang, Lesion mask-based simultaneous synthesis of anatomic and molecular mr images using a gan, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2020, pp. 104–113.

[191] T.C. Mok, A. Chung, Learning data augmentation for brain tumor segmentation with coarse-to-fine generative adversarial networks, in: International MICCAI Brainlesion Workshop, Springer, 2018, pp. 70–80.

[192] H. Uzunova, J. Ehrhardt, H. Handels, Generation of annotated brain tumor MRIs with tumor-induced tissue deformations for training and assessment of neural networks, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2020, pp. 501–511.

[193] A. Segato, V. Corbetta, M. Di Marzo, L. Pozzi, E. De Momi, Data augmentation of 3D brain environment using deep convolutional refined auto-encoding alpha GAN, IEEE Trans. Med. Robot. Bionics 3 (1) (2020) 269–272.

[194] T. Kossen, P. Subramaniam, V.I. Madai, A. Hennemuth, K. Hildebrand, A. Hilbert, J. Sobesky, M. Livne, I. Galinovic, A.A. Khalil, et al., Synthesizing anonymized and labeled TOF-MRA patches for brain vessel segmentation using generative adversarial networks, Comput. Biol. Med. 131 (2021) 104254.

[195] T. Xia, A. Chartsias, C. Wang, S.A. Tsaftaris, A.D.N. Initiative, et al., Learning to synthesise the ageing brain without longitudinal data, Med. Image Anal. 73 (2021) 102169.

[196] Y. Chen, X.-H. Yang, Z. Wei, A.A. Heidari, N. Zheng, Z. Li, H. Chen, H. Hu, Q. Zhou, Q. Guan, Generative adversarial networks in medical image augmentation: a review, Comput. Biol. Med. (2022) 105382.

[197] M. Li, G. Zhou, A. Chen, J. Yi, C. Lu, M. He, Y. Hu, FWDGAN-based data augmentation for tomato leaf disease identification, Comput. Electron. Agric. 194 (2022) 106779.

[198] M. Xu, S. Yoon, A. Fuentes, J. Yang, D.S. Park, Style-consistent image translation: A novel data augmentation paradigm to improve plant disease recognition, Front. Plant Sci. 12 (2021) 773142.

[199] H. Jin, Y. Li, J. Qi, J. Feng, D. Tian, W. Mu, GrapeGAN: Unsupervised image enhancement for improved grape leaf disease recognition, Comput. Electron. Agric. 198 (2022) 107055.

[200] Y. Jing, Y. Bian, Z. Hu, L. Wang, X.-Q.S. Xie, Deep learning for drug design: an artificial intelligence paradigm for drug discovery in the big data era, AAPS J. 20 (3) (2018) 1–10.

[201] D. Dana, S.V. Gadhiya, L.G. St. Surin, D. Li, F. Naaz, Q. Ali, L. Paka, M.A. Yamin, M. Narayan, I.D. Goldberg, et al., Deep learning in drug discovery and medicine; scratching the surface, Molecules 23 (9) (2018) 2384.

[202] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov, A. Zhavoronkov, The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology, Oncotarget 8 (7) (2017) 10883.

[203] A. Kadurin, S. Nikolenko, K. Khrabrov, A. Aliper, A. Zhavoronkov, druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico, Mol. Pharmaceut. 14 (9) (2017) 3098–3104.

[204] G.R. Padalkar, S.D. Patil, M.M. Hegadi, N.K. Jaybhaye, Drug discovery using generative adversarial network with reinforcement learning, in: 2021 International Conference on Computer Communication and Informatics, ICCCI, IEEE, 2021, pp. 1–3.

[205] D. Manu, Y. Sheng, J. Yang, J. Deng, T. Geng, A. Li, C. Ding, W. Jiang, L. Yang, FL-DISCO: Federated generative adversarial network for graph-based molecule drug discovery: Special session paper, in: 2021 IEEE/ACM International Conference on Computer Aided Design, ICCAD, IEEE, 2021, pp. 1–7.

[206] J. Konečný, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, 2016, arXiv preprint arXiv:1610.05492.

[207] P. Dhariwal, A. Nichol, Diffusion models beat gans on image synthesis, Adv. Neural Inf. Process. Syst. 34 (2021) 8780–8794.

[208] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Adv. Neural Inf. Process. Syst. 33 (2020) 6840–6851.

[209] Y. Song, S. Ermon, Generative modeling by estimating gradients of the data distribution, Adv. Neural Inf. Process. Syst. 32 (2019).

[210] F.-A. Croitoru, V. Hondru, R.T. Ionescu, M. Shah, Diffusion models in vision: A survey, 2022, arXiv preprint arXiv:2209.04747.

[211] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, M. Norouzi, Palette: Image-to-image diffusion models, in: ACM SIGGRAPH 2022 Conference Proceedings, 2022, pp. 1–10.

[212] Y. Jiang, S. Chang, Z. Wang, Transgan: Two transformers can make one strong gan, 2021, arXiv preprint arXiv:2102.07074 1, 3.

[213] Z. Lv, X. Huang, W. Cao, An improved GAN with transformers for pedestrian trajectory prediction models, Int. J. Intell. Syst. 37 (8) (2022) 4417–4436.