Advances of Generative Adversarial Networks: A Survey

Dirk Hölscher*, Christoph Reich*, Martin Knahl*, Frank Gut* and Nathan Clarke†
*Institute for Data Science, Cloud Computing and IT Security, Furtwangen University, 78120 Furtwangen, Germany
Email: {dirk.hoelscher, christoph.reich, martin.knahl, frank.gut}@hs-furtwangen.de
† Centre for Security, Communications and Network Research, Plymouth University, Plymouth, U.K.
Email: n.clarke@plymouth.ac.uk

***Abstract*—Generative Adversarial Networks (GANs) are part of the deep generative model family and able to generate synthetic samples based on the underlying distribution of real-world data. With expanding interest new discoveries and recent advances are hard to follow. Recent advancements to stabilize training, will help GANs to open up new domains using adjusted architectures and loss functions. Various findings show, that GANS can be used to generate not only images, but is also useful for text and audio creation. This paper, presents an overview of different GAN architectures, giving summaries of the underlying fundamentals of each presented GAN. Furthermore, this paper presents look into four application domains and lists additional domains. Additionally, this paper summaries datasets and metrics used to evaluate GANs and present recent scientific advancements.**

Keywords–generative adversarial networks; machine learning; deep learning.

I. Introduction

Generative Adversarial Networks (GANs) have revolutionized deep generative models used to learn data distributions with unsupervised learning, to generate synthesized samples for various domains coming from a number of sources resembling the true data distribution. Introduced by Goodfellow et al. [1] in 2014 it has been the focus of many researchers to apply the concept to new domains, introduce new loss functions and architectures, and new approaches to stabilize the training process. Applicability of GANs is growing rapidly and reaches new domains while achieving better results by applying adapted architectures and joining domain typical methods. The shared goal of all GANs is to reach a Nash Equilibrium [2], meaning that neither the discriminator nor the generator can be further improved. The idea of GANs originates from the game theory, a classical two-player zero-sum game [3], with only one winner. Nowadays, GANs are not only used to create synthesized images [4]-[5], but they can also create text [6], perform image and video translations [7]-[8], video summaries [9], copy objects into another image [10], help to reconstruct archaeological findings [11] or create images from text descriptions [12]. Each of these GANs is based on a different architecture with varying loss functions. All these variations and specialized architectures combined with missing unified evaluation methods and datasets [13] make it difficult to compare and evaluate the performance of GANs. Therefore, it is important to find available and commonly used metrics and datasets used in the scientific community to be able to compare the different approaches and underlying architectures regarding their performance. This survey gives an overview for researchers and summarizes the current state-of-the-art of GANs based on six research questions about architectures, domains, evaluation (metrics and datasets), prevailing problems and advances in research.

The paper is organized as follows: Section II introduces the methodology used for this survey and gives a short overview about publications. In Section III, common and novel GAN architectures are listed and summarized. Afterwards, Section IV gives a short overview and describes research done in four domains, where different styles of GANs are deployed. Section V summarizes datasets used for evaluation by the previous described GAN architectures and lists GAN evaluation methods. Furthermore, a database search was conducted to find the most prominent evaluation methods. Section VI takes a look at occurring problems while training GANs and shows advances in research. SectionVII concludes this paper.

II. Methodology and Overview

The following section will detail the methodology used to create the survey and shows the process how contributions were searched and selected. Furthermore, this section gives an overview about publications distribution in recent years.

A. Research Questions

The Research Questions (RQ) regarding GANs are as follows:

- RQ1: What GAN architectures exist?
- RQ2: In what domains are GANs utilized?
- RQ3: What datasets are used to evaluate GAN performance?
- RQ4: What metrics are deployed to validate GANs?
- RQ5: What challenges exist when working with GANs?
- RQ6: Advances in research?

All subsequent Sections are structured to answer the above mentioned RQs.

B. Database Search

The search was conducted using the following four databases:

- IEEE Xplore
- ACM Digital Library
- NIPS Proceedings
- arXiv (used to find additional publications in the first three mentioned databases)

Search terms were focused on clear easy structures. A general search without using any specialized search string yields a result of more than 430,000 results for IEEE Xplore and ACM alone. By searching titles, full text and abstracts about the full term and synonym achieved the best results. Searching was done with the following search term: *"Publication Title": "generative adversarial network*" OR "gan" AND "Abstract": "generative adversarial network*"*. With gaining

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-793-1

popularity of GAN, more publications starting to appear as shown in Figure 1.

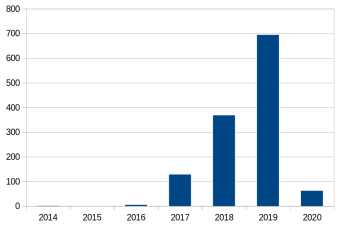

Figure 1. Annualised Number of publications

A big chunk of publications were published in conference proceedings (89%) while only 11% were journals. There is a continuous rise in publications following 2014 with 2015 being an outlier for the searched databases (arXiv excluded, due to limited filtering options and inclusion of only peer-reviewed publications). At the time of this survey 2020 had 62 publications, excluding early access titles. It can be assumed that publications will continue to increase in the coming years.

III. Generative Adversarial Network Architectures

With the introduction of GANs by Goodfellow et al. [1] in 2014, various new adapted architectures addressing a variety of problem domains, were introduced in the following years. This section will list and describe some of the most prominent and concept wise interesting GAN architectures and is related to RQ1. The different architectural patterns are sorted by their date of publication to indicate the growth and progress in recent years. Figure 2 shows a generic GAN architecture with all components, as well as the input for Generator(G) and Discriminator(D). G takes a random noise vector as input and generates a fake sample forwarded to D as input. D know the real data distribution, and classifies the input either as being real or fake and backpropagates the error.

Figure 2. Generic GAN Architecture

A. Generative Adversarial Nets

The first GAN introduced by Goodfellow et al. in 2014 [1], introduced the concept of GAN's as a two-player minimax game where two neural networks G and D compete against each other. The generator takes random noise as input and tries to generate an output consistent with the original data. The discriminator tries to classify if the input came from the original data or from the generator. This can be images for example. The goal of D is to maximize the probability to distinguish samples that came from the original data and sample generated by G. At the same time G is trained to minimize the probability of getting caught by D, meaning that generated samples forwarded to D are classified as part of the original data. The equation of the minimax game is given with the following function V(G,D) also known as a vanilla GAN:

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

Where x are the data real instances, $E_{x\sim p_{data}(x)}$ is the expected value over all x $D(x)$ is D's probability estimation that an instance from x is real, z is the noise input vector, $G(z)$ is the G's generated output using z, $E_{z\sim p_z(z)}$ is the expected value over all generated fake samples and $D(G(z))$ is the probability that a fake instance is classified as real.

B. Conditional Generative Adversarial Nets

The cGAN by Mirza and Osindero [4] is a extension of the vanilla GAN, where both D and G are conditioned using additional information, for example class labels. This can be achieved by adding another input layer for D and G that contains the conditioned information. The input noise from the vanilla GAN is combined with additional information and combined to form a joint hidden representation (e.g., a multilayer perceptron (MLP) hidden layer). The discriminator takes the original data as well as the conditioned information as input for a discriminative function. This minimax game can be represented as the following equation given by the authors:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y)))] \quad (2)$$

The equation of the minimax game is the same as (1) with the additional of the condition y to the discriminative function. Figure 3 illustrates the structure of a cGAN with the conditioned input. where the blue circles are the random noise z and green is the conditioned information y fed into the layers of the network. Both z and y are G's input, where y acts an restriction for instance creation. D utilizes y to determine if a sample is real or fake and knows the real data instances as well as the condition y.

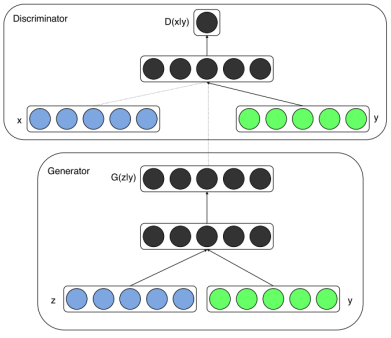

Figure 3. cGAN Architecture [4]

C. InfoGAN

InfoGAN proposed by Chen and Deng [14] in 2016 splits the input random noise vector into two vectors. This is done to make the GAN learn meaningful representations. The first vector contains incompressible noise(z), while the second vector (latent code) targets the data's semantic features(c). G takes both vectors as input and is represented as *G(z,c)*. To prevent trivial codes, where the generator ignores c to satisfy:$P_G(x|c) = P_G(x)$, meaning G is able to ignore the

latent vector to generate samples. Therefore, an information-theoretic regularization is introduced stating that between *c* and *G(z,c)* the amount of mutual information must be high. Based on information theory, mutual information is the measurement of learned knowledge between two random variables *Y* and *X*. This can be expressed as the difference between two entropies (X and Y) with their mutual information (I):

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \quad (3)$$

This ensures that the information in c is not lost in G. This leads to an information-regularized minimax game with the following equation:

$$\min_G \max_D V_I(D,G,Q) = V(D,G) - \lambda I(c;G(z,c)) \quad (4)$$

The equation is a modified version of (1) with λ added as a mere factor, to ensure the use of latent code in the network. A mutual information term can be hard to maximize but is achievable by utilizing a lower bound auxiliary distribution *Q(c|x)* known as Variational Information Maximization [15]. The auxiliary distribution *Q* is implemented as a neural network. D and Q share every convolutional layer with one fully connected one, which is responsible for the parameters of the conditional distribution *Q(c|x)*. If *c* is of categorical nature, softmax is used to represent Q, while continuous *c* is dependent on the true posterior.

D. Wasserstein GAN

The Wasserstein GAN (WGAN) by Arjovsky, Chintala and Bottou [16] is an extension of the vanilla GAN with benefits of improved stability and a loss function correlating with G's convergence. WGAN updates D more often for each training iteration than it updates G. WGAN utilizes an approximation of the Earth-Mover (EM) distance or Wasserstein metric [17] to establish above mentioned benefits when compared with vanilla GANs. The EM distance as shown in (5) is the optimal cost of transporting an amount of mass from *x* to *y* to transform $\mathbb{P}_r$ into $\mathbb{P}_g$. Meaning the infimum or cheapest cost for any transport with γ being all calculated plans and $W(\mathbb{P}_r, \mathbb{P}_g)$ being the probability distribution of x and y.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|] \quad (5)$$

Therefore, a function f solving the maximization problem of (6) must be found.

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)] \quad (6)$$

Sup is the least upper bound and symbolized by f is 1-Lipschitz function [18]. To do this, a parameterized neural network with compact space (W) and weights w is trained with backpropagation. This means all functions of w will be K-Lipschitz. This is achieved by fitting the weights into a fixed value range after each gradient update. The Wasserstein distance can be calculated with the found and learned 1-Lipschitz function.

E. Deep Convolutional Generative Adversarial Network

In 2015 Radford, Metz and Chintala [19] introduced DCGANs. One way to use DCGANs is by learning representations of unlabeled images to use them in supervised learning. Such a feat can be achieved by using parts of D and G for feature extraction in supervised tasks. DCGANs combine the idea of connecting GANs and Convolutional Neural Networks (CNN) by incorporating architectural changes of CNNs. The first change made was the use of all convolutional networks without deterministic spatial functions and allowing the learning of spatial downsampling by using strided convolutions. G profits from spatial downsampling and additionally learns its own discriminator. The second is the elimination of fully connected layers. Instead, DCGANs utilize global average pooling as an trade-off between training stability and model convergence. G has one connected layer which is the input of the noise vector shaped into a 4-dimensional tensor used to build the convolution stack. D's last layer is flattened and fed through a single sigmoid output. The next change pertains batch normalization, helping to prevent some occurring training problems in deeper models due to poor initialization. Furthermore, the addition of batch normalization helped deep generators to learn and prevented collapse on a single point. Every layer is fitted with batch normalization expect the generator's output and the discriminator's input layer. All layers of G use ReLU (Rectified linear unit - a linear function for values greater than zero and non-linear for negative values) as activation function except the output layer which uses Tanh (hyperbolic tangent - is a nonlinear function between $-1, 1$, averaging around zero). For D leaky rectified is used for activation. Figure 4 shows the generator's architecture which takes a 100x1 noise vector (z) and runs it through the generator to map it into a 64x64x3 output $G(Z)$. The input noise is reshaped and expanded into a 4-dimensional tensor to form a vector of size 1024x4x4.

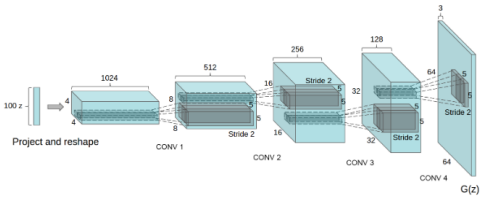

Figure 4. DCGAN Architecture [19]

An extension to DCGAN is called Regularized DCGAN (RDCGAN) by Mehralian and Karasfi [20]. Representation learning is as good as DCGANs ability but prevents mode collapse (see VI-A) with the help of an encoder (E) with the benefit of i) stabilize training and ii) provide more features. Based on DCGAN, RDCGAN is an unsupervised method that can learn good representations of images and by training through adversarial learning the discriminator and encoder can be reused for image classification tasks. DCGAN's structure persists with an added encoder to learn image encodings and feed them to the generator for input reproduction. To show the efficiency of RDCGAN's classification abilities, a classifier is situated at the top of the network. As with vanilla GAN D is trained to maximize $D(x)$ to identify images from the original data, while G is minimizing $D(G(z))$ for getting caught with fake images. While training G, D is frozen and the the encoder is only frozen while G is training to create fake images. The goal of this approach is to generate fake images with noise to fool D and reconstruct encoded images from the input. Therefore, G has two loss functions to enable G not only to generate images but to have the ability to reconstruct images. With l_{con}a reconstruction loss function the distance between real and generated image from E can be minimized and is

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-793-1

depicted in the following equation:

$$\mathrm{l}_{con} = \mathbb{E}_{x \sim P_{data}(x)} \|x - G(E(x))\|_2 \tag{7}$$

To enforce G to reconstruct images which will pass through Ds tests the encoder is defined as follows:

$$\mathrm{l}_{encoder} = \mathbb{E}_{x \sim P_{data}(x)} \left(\log(D(G(x))) + \varepsilon\right) + \mathrm{I}_{con}) \tag{8}$$

By minimizing E, it can extract more useful features for G to reconstruct the image and D will classify them as real images with a higher chance. For G to fool D, G has the following object function trying to minimize $\log(\mathrm{D}(\mathrm{G}(\mathrm{z})))$:

$$\mathrm{l}_{generator} = \mathbb{E}_{z \sim P_z(z)} \left(\log(D(G(z))) + \varepsilon\right) + \mathrm{I}_{encoder}) \tag{9}$$

F. Image-to-Image Translation

In [7], Zhu, Park, Isola and Efros, present an unpaired Image-to-Image translation architecture called CycleGAN. Paired Image-to-Image translation ($\{x_i, y_i\}_{i=1}^{N}$) means for each x_i there is an corresponding y_i and vice versa. Unpaired translations differ, because there is no information given which x_i matches y_i. Therefore, unpaired approaches have a source $\{x_i\}_{i=1}^{N} \in X$ and specify a target set for the translation $\{y_i\}_{i=1}^{M} \in Y$. The idea of cycle consistency is that $G : X \rightarrow Y$ and $F : Y \rightarrow X$ are consistent meaning that the translation of an image from the source X to the target Y and back will yield the starting image from source X again. Forward cycle consistency ($x \rightarrow G(x) \rightarrow F(G(x)) \approx x$) is the ability to translate each image from X back to its original state. The same is true for each image from Y, called backward cycle consistency ($y \rightarrow F(y) \rightarrow G(F(y)) \approx y$). This results in the following equation:

$$\begin{aligned}\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)} \left[\|F(G(x)) - x\|_1\right] \\ + \mathbb{E}_{y \sim p_{data}(y)} \left[\|G(F(y)) - y\|_1\right]\end{aligned} \tag{10}$$

Two discriminators(D_X and D_Y) are used to differentiate images $\{x\}$ and translated images $\{F(y)\}$ for D_X and $\{y\}$, $\{G(x)\}$ for D_Y. The adversarial loss function for D_Y is:

$$\begin{aligned}\mathcal{L}_{\mathrm{GAN}}\left(G, D_Y, X, Y\right) = \mathbb{E}_{y \sim p_{data}(y)} \left[\log D_Y(y)\right] \\ + \mathbb{E}_{x \sim p_{data}(x)} \left[\log\left(1 - D_Y(G(x))\right)\right]\end{aligned} \tag{11}$$

The complete equation for the specified problem is as follows:

$$G^*, F^* = \arg\min_{G,F} \max_{D_x, D_Y} \mathcal{L}\left(G, F, D_X, D_Y\right) \tag{12}$$

CycleGAN architecture is based on the neural style transfer in [21]. The network structure contains two stride-2 and two $\frac{1}{2}$-strides convolutions, as well as residual blocks. D is using PatchGANs with a patch size 70x70 to classify if these patches are real or were generated by G. To stabilize the training the negative log likelihood in (12) is replaced by least square loss and results in the following equation:

$$\begin{aligned}\mathcal{L}_{\mathrm{LSGAN}}\left(G, D_Y, X, Y\right) = \mathbb{E}_{y \sim p_{data}(y)} \left[\left(D_Y(y) - 1\right)^2\right] \\ + \mathbb{E}_{x \sim p_{data}(x)} \left[D_Y(G(x))^2\right]\end{aligned} \tag{13}$$

For the occurring problem of model oscillation both discriminators are updated with a history of previous generated images instead of using the newest generated images.

In contrast, the Image-to-Image translation architecture presented by Isola et al. in [22], is a paired approach. In this case, G must not only fool D but must also achieve near ground truth output. Therefore, L1 distance is chosen:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z} \left[\|y - G(x, z)\|_1\right] \tag{14}$$

This leads to the objective's equation of:

$$G^* = \arg\min_{G} \max_{D} \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{15}$$

The architecture is based on [19] (see Section III-E)). To prevent some bottlenecks G is U-Net [23] shaped with connections between each layer i and $n - i$ with n being the overall quantity of layers. All channels of i are connected with the channels of $n - i$. Their used D is called PatchGAN also used in the above CycleGAN and tries to classify a patch of size NxN in an image of being real of fake. D runs this patch across the whole image averaging the results of all patches to generate the output of D.

G. StyleGAN

StyleGAN by Karras, Laine and Aila [5] is an enhancement from their previous published ProGAN [24]. StyleGAN proposes a new generator by including style transfer ideas. G starts learn from a continuous input and at each convolution layer, adjusts the style based on latent code (see Section III-C). In combination with noise the network can distinguish and separate attributes and random variations. Instead of starting from the input layer, StyleGAN's architecture starts from a learned constant. The latent code (z) is embedded into a latent space Z. Instead of putting z directly into G's input layer, a non-linear mapping is used to map the input into an intermediate latent space W ($f : \mathcal{Z} \rightarrow \mathcal{W}$)G is controlled by the latent space through adaptive instance normalization at each layer. StyleGANs architecture is shown in Figure 5

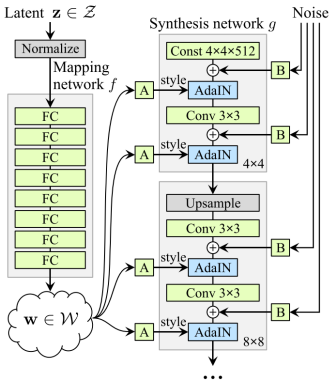

Figure 5. StyleGAN Generator Architecture [5]

and shows a progressively growing G. The network starts with 4x4 images and are trained until stable and then the size is doubled. The network consists of 8 layers while the synthesis network g consists of 18 layers, each resolution has two layers till 1024^2 is reached. The last layer's output is converted into RGB by using a 1x1 convolution. The depicted mapping network takes a random point from latent space and creates a style vector which is transformed and utilized after each convolution. Adaptive instance normalization (AdaIN) standardizes the the output of feature maps to a standard Gaussian and adds the style vector as bias. Additional noise is added for style variations of the generated images. StyleGAN enables image synthesis control with the help of scale-specific

modifications for each style. A style is drawn from each learned distribution and based on the collected styles the synthesis network can generate new images. Style mixing is used to use more than one latent space to generate a defined percentage of images during training. While such an image is generated at some random point one latent space is switched for another.

H. StackGAN

StackGAN [25] and their enhancement StackGAN+++ by Zhang et al. [12] is a stacked GAN approach where GANs are stacked in a tree-like shape. StackGAN-v1 is used for text-to-image synthesis. Therefore, a text description is encoded resulting in a text embedding φ_t. This first GAN is generating low resolution images based on φ_t focusing on rough outlines and object colors. Gaussian conditioning variables $\hat{c}_0$ are sampled from a conditioning augmentation to create more training pairs from a Gaussian distribution ($\mathcal{N}\left(\mu_0\left(\varphi_t\right), \Sigma_0\left(\varphi_t\right)\right)$). A random variable z is used to train the generator (G_0) and discriminator (D_0) shown in the following equations:

$$\begin{aligned}\mathcal{L}_{D_0} = & \mathbb{E}_{(I_0,t)\sim p_{data}}\left[\log D_0\left(I_0, \varphi_t\right)\right] \\ & + \mathbb{E}_{z\sim p_z, t\sim p_{data}}\left[\log\left(1 - D_0\left(G_0\left(z, \hat{c}_0\right), \varphi_t\right)\right)\right]\end{aligned} \tag{16}$$

$$\begin{aligned}\mathcal{L}_{G_0} = & \mathbb{E}_{z\sim p_z, t\sim p}\left[-\log D_0\left(G_0\left(z, \hat{c}_0\right), \varphi_t\right)\right] \\ & + \lambda D_{KL}\left(\mathcal{N}\left(\mu_0\left(\varphi_t\right), \Sigma_0\left(\varphi_t\right)\right) \| \mathcal{N}(0, I)\right)\end{aligned} \tag{17}$$

The goal is to maximize L_{D_0} and minimize L_{G_0}. I_0 is the real image and t the original description form p_{data}, z is the noise vector and λ is used a regulation parameter to balance the terms in (17), set to 1 for the tests conducted in the paper. A pre-trained character level CNN-RNN text encoder is used to map text descriptions to the feature space of images learning a correspondence between images and descriptions. First φ_t is fed into a fully connected layer to generate the necessary parts for the shown Gaussian distribution to sample $\hat{c}_0$. The discriminator compresses the text embedding into N_d dimensions with a fully connected layer and then used to form a $M_d \times M_d \times N_d$ tensor. The image gets downsampled until it reaches $M_d \times M_d$ dimensions and is linked with the text tensor. The result is forwarded to a 1×1 convolutional layer to learn features of the text and image. The decision score is made by single node fully connected layer.

Stage-II GAN is built on top of Stage-I GAN and tries to generate high-resolution images. It is conditioned on images with low resolutions and as well as text embedding to correct previous mistakes made by Stage-I GAN. The low-resolution images of G_0, resulting in $s_0 = G_0\left(z, \hat{c}_0\right)$, and the Gaussian latent variables $\hat{c}$ are used to train D and G of the Stage-II GAN by maximizing D and minimizing G as shown in the following equations:

$$\begin{aligned}\mathcal{L}_D = & \mathbb{E}_{(I,t)\sim p_{data}}\left[\log D\left(I, \varphi_t\right)\right] \\ & + \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{data}}\left[\log\left(1 - D\left(G\left(s_0, \hat{c}\right), \varphi_t\right)\right)\right]\end{aligned} \tag{18}$$

$$\begin{aligned}\mathcal{L}_G = & \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{data}}\left[-\log D\left(G\left(s_0, \hat{c}\right), \varphi_t\right)\right] \\ & + \lambda D_{KL}\left(\mathcal{N}\left(\mu\left(\varphi_t\right), \Sigma\left(\varphi_t\right)\right) \| \mathcal{N}(0, I)\right)\end{aligned} \tag{19}$$

During this stage the Stage-II GAN is not fed with additional random noise, as the occurrence of randomness should have already happened and been stored in s_0. Gaussian variables from both stages have the same pre-trained encoder and generate the same φ_t. The main difference are different fully connected layers for condition augmentation to generate differing deviations and means. Stage-II GAN is able to learn new information which were previously omitted by Stage-I GAN. G of the Stage-II GAN is constructed as an encoder-decoder network using residual blocks. As with Stage-I generator φ_t is used to generate $\hat{c}$. The results s_0 of the first GAN are is downsampled until it reaches a size of $M_g \times M_g$. All encoded features are fed into residual blocks, able to learn multi-modal representations. In the last step a decoder is used to generate a high-resolution image. The discriminator receives an extra downsampling block to handle the larger image size. For training, D works with positive samples of real images and their textual descriptions and negative samples consisting of real images and mismatched textual descriptions and generated images with their textual description.

Upsampling is done by nearest-neighbor block followed by a stride-1 convolution with 3×3. Each layer except the final layer apply batch normalization and ReLU activation function. Each residual block consists of stride-1 convolution with 3×3, batch normalization and ReLU activation. Two of these blocks are used in Stack-I GAN with a 128×128 resolution and is doubled for 256×256 models. Downsampling layers, with the exception of the first block, have Batch normalization, a LeakyReLU activation and stride-2 convolution with 4×4.

I. Copy-Pasting GAN

Arandjelovic and Zisserman introduced a Copy-Pasting GAN in [10]. The approach is about unsupervised object discovery, where G learns how to discover objects and pasting it into another so that D will be fooled. The main difference with other GAN and object discovery methods is G, which does not generate objects and its solely responsibility is to detect and segment existing objects. Instead of creating a whole new image G combines two images by selecting and copying a section of the source image to the destination image. Therefore, the generator is restricted. A convolutional network takes the source image (I_s) to generate a segmentation mask. A new image is generated by copying and blending (I_s) into the new image (I_d) using the copy mask ($m_\theta(I_s)$) with values of m are in $[0, 1]$, θ symbolizing trainable parameters and $\odot$ being the element-wise product, resulting in the following equation:

$$\begin{aligned}I_c &= C\left(I_s, I_d, m_\theta\left(I_s\right)\right) \\ &= m_\theta\left(I_s\right) \odot I_s + \left(1 - m_\theta\left(I_s\right)\right) \odot I_d\end{aligned} \tag{20}$$

The same masked is used across all channels. This applies a limit to the copying and pasting operation of G by forcing it to paste the object at the same location without any transformation. G is implemented as a U-Net with a 1-channel output, resulting in ($m_0(I_s)$). D is also a U-Net with an average-pooled encoding in the middle and passed on to fully connected layer for classification.

IV. GAN Application Domains

The field of application for GANs is constantly growing and finds its way into various domains. The following section, will list additional GAN architectures extending the concept of the architectures presented in Section III and list them based on their domain together with a short summary and if applicable the architectural type from Section III. This section extends the findings of RQ1 from the previous section and introduces

exemplary domains to show the variety of GANs. Furthermore, the domains in this section answers RQ2.

A. Autonomous Driving

Autonomous driving is complex domain with various tasks and complex image structures depicting various objects of different types and distances. There are several different use cases for autonomous driving where GANs can help to depict situations that are highly unlikely and difficult to capture (e.g., pedestrian crossing the street relatively close to the vehicle or even getting by it). In [26] Choi, Jeong, Park and Ha, propose an image generation GAN based on DCGAN (see Section III-E) to create new situational images with the help of feature maps. This approach extracts a feature map (e.g., of pedestrians) and places them in another scene. A more specialized approach is presented in [27] placing pedestrians with different poses into preexisting scenes by using an encoder to extract a person from a scene and given to mask estimation network deciding which pixel should be taken and insert into the original image to place pedestrians. In [28], GANs are used for augmentation in the domain of autonomous driving vehicle. First a basic introduction about GANs is given. Afterwards, the paper introduces a CycleGAN (see III-F) experiment is conducted where the condition of front mounted camera is changed from soiled to clean and vice versa. In [29] the authors present DeepRoad a validation framework for autonomous driving systems. The framework can generate new images with various weather conditions based on real-world weather scenes. Using synthetic images the consistency of the system can be tested, and validates images for DNNs with their VGGNet features by measuring the distance between input and training images and is closely related to CycleGAN (see III-F).

B. Archaeology

In [11] Hermoza nad Sipiran introduce ORGAN a 3D reconstruction GAN to restore archaeological objects. ORGAN is based on an encoder-decoder 3D DNN on top of a GAN based on cGANS (Section III-B). With two loss functions (completion loss and Wasserstein loss) the network is trained to predict the missing parts of a damaged object. To compensate differences between archaeological objects, the cGAN is conditioned on variable containing information about the culture, region or the object itself. ORGAN is able to reconstruct objects with nearly 50% missing. Another approach is the 3D reconstruction of objects from single photographs by Kniaz, Remondino and Knyaz [30]. They use a Z-GAN (based on pix2pix, Section III-F) for image-to-voxel translation. The generator's encoder is unchanged while the 2D kernels are changed into 3D convolutional kernels. Like a U-Net, Z-GAN has skip connections (in accordance with U-Net definition) helping to transfer high-frequency components of the input to the 3D shape. The generated reconstructions are reviewed and compared with domain experts and shows the strength of the approach.

C. Video Editing

In [9], Zhang, Zhao, Kampffmeyer and Tan, propose DTR-GAN a dilated temporal relational GAN for video summarization. DTR-GAN allows a frame-level summarization without loosing viable information within the video. Therefore, a DTR module is constructed capturing relations between neighbouring multi-range frames. One DTR layer consists of four DTR units of different size to capture the relations between frames. G tries to learn to identify key frames. A Temporal Encoding Module integrates Bi-LSTM (Long Short Term Memory) layer as well as a three layer DTR network. A confidence score is predicted for key frame capturing confidence. D learns the relations between original video and summary using a three-player loss to ensure that D can recognize and learn the difference between valid summaries and trivial ones. Mocycle-GAN [8] proposed by Chen, Pan, Yao, Tian and Mei, is an unpaired video-to-video translation process comparable with image-to-image translations (Section III-F). Mocycle-GAN transfers videos from a source domain to a target domain with two generators G_X and G_Y for cross domain frame synthesis, two discriminators (D_X and D_Y) to distinguish between real and synthetic frames and a motion translator (M_X) for cross domain motion translation. Real frames are first translated into synthetic frames with G_X and further transformed via inverse mapping of G_Y into reconstructed frames. Furthermore, a FlowNet [31] is used to obtain optical flows for motion representation before and after forward cycles. While training Mocycle-GAN utilizes three temporal constraints for structural appearance and temporal continuity exploration. Adversarial learning using adversarial constraints ensures realistic synthetic frames, frame and motion cycle consistent constraint is used for inverse translations for frames and motions while the third motion translation constraint validates cross domain motions transfers. DCVGAN [32] propose an advanced video generating GAN using optical information as well as 3D geometrical information with depth video. The generator is based on MoCoGAN [33] and splits the latent space into content and motion. A RNN models and generates video dynamics and motion latent vectors, while a CNN is responsible for depth image generation from latent vectors. Colour to depth translation is achieved by using a pix2pix(Section III-F) based architecture. Using U-Net structures the colour image generator is based on an encoder-decoder architecture. Two discriminators one for images, using randomly selected colour and depth images to test if it is an original sample or generated, and one for videos taking the same input as video and evaluating its temporal features. With the addition of depth images DCVGAN can outperform MoCoGAN.

D. Medicine

In [34] the authors propose a GAN for treatment recommendations based on patient-centric literature retrieval. The goal of the approach is to measure the similarity of gene mutations. Therefore, the input is encoded as two-hot vectors with the number of genes used as vector dimensions. G outputs one entire batch per training step while D receives one sample and the batch average. Furthermore, D is parallel trained with an online training scheme receiving only a single sample and a batch average of zero with a training mode indication. A metric is used to rank patient document pairs, by looking for genes mentioned in both records (patient and documents). To gain treatment adequacy the authors separate D and G and condition both on the patient information vector. G is trained as a feed-forward network to predict patient treatments and D is used to distinguish between real and synthetic vectors. Scores are calculated for D and G by taking the conditioned output of D and G and comparing it with the treatment vector by applying cosine similarity. Both scores are fused to obtain the final document score. In [35], Xie, Xu and Li, propose

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-793-1

a GAN for CT reconstruction of images taken from limited angles with artefacts. The proposed GAN is based on cGAN (Section III-B) and uses Wasserstein distance (Section III-D) for training stability. The proposed GAN uses perceptual loss as well as adversarial loss as loss functions. The proposed approach shows that it can remove artefacts while retaining the CT's texture details.

E. Additional GAN Domains

The applicability of GANs is not limited to the four previous shown domains. In [36], Wen, Singh and Raj, proposes a framework to reconstruct faces based on their voices by matching identities between generated faces and speakers. Engel, Agrawal, Chen, Gulrajani, Donahue and Roberts [37] generated high quality audio samples using GANs with latent and pitch vectors with global conditioning instead of WaveNet and rapidly sped up the process. The importance of GANs for augmentation are presented and discussed in [38]. In [6] an actor-critic conditional GAN is presented. The GAN can learn surrounding context from text to fill in blanks. Parts of the text are deleted and the model tries to fill the missing pieces to look like the original. ChinaStyle[39] introduces a new dataset set with six categories and 1913 total images about Chinese traditional figure painting. Furthermore, the authors introduce MA-GAN a style transfer GAN for image translation of portraits into Chinese paintings. A more exotic domain can be found in [40]. RamenGAN is a cGAN using an additional discriminator to create ramen with rounder dishes. The other proposed RecipeGAN a WassersteinGAN is used to receive ingredients of a dish via image search.

V. GAN Evaluation Methods

This section, will list and highlight common datasets used to evaluate new approaches and will further categorize all presented GANs from Section III. Furthermore, this section, will show and introduce commonly used metrics to evaluate the performance of GANs. All models are generative ones but differ in their approaches and thus use different datasets for evaluation. Categorizing by datasets is one possible way to filter GANs. Furthermore, this section, will present GAN evaluation measures.

A. Datasets

Table I summarizes datasets used by well known GAN architectures for image generation. The next architectural style are Style-Transfer GANs as summarised in table II which maps the properties from one domain to another. For example, this includes the transfer from day to night or hand drawing to real image.

TABLE II. DATASET OVERVIEW STYLE-TRANSFER GANs

Architecture	Cityscape	CMP Facades	ImageNet
CycleGAN	X	X	X
Pix2Pix	X	X	X

The final Table III summarizes specialized architectures for text-to-image processing and image manipulation.
The variance of used datasets is quite vast and limits model comparison immensely. As shown above based solely on literature, a comparison between some models is possible, but cannot be guaranteed. For this reason, Lucic, Kurach, Michalski, Bousquet and Gelly [13] propose a simple dataset for model evaluation. The above shown and currently used datasets are either too simple or complex to achieve meaningful evaluation results. Therefore, a data manifold is created which allows efficient computation between sample distance to manifold. The problem is transformed into a computational problem, where the precision is higher the closer the samples from model distribution are to the manifold. High recall in this case means that G is able to generate samples relatively close to the manifold. The proposed manifold is of convex polygons. For single-channel grey images ($x \in \mathbb{R}^{d^2}$) the sample distance to the manifold is the squared Euclidean distance from $\hat{x} \in \mathbb{R}^{d^2}$ to C_3, the manifold's closest sample, as shown in the following equation:

$$\min_{x \in C_3} \ell(x, \hat{x}) = \sum_{i=1}^{d^2} \|x_i - \hat{x}_i\|_2^2 \tag{21}$$

The approximate of a solution is found by gradient descent on the convex polygon's vertices, with each iteration being a valid convex polygon. Using random initial solutions the algorithm is executed 5 times to minimize false-negative samples. The findings of RQ3 are summarized in the tables of this section and give an overview of the different GAN architectures and their datasets utilized for evaluation.

B. GAN Evaluation

This subsection, will highlight the most important metrics and measurements used to evaluate GANs. The results are based on the conducted literature review performed to answer RQ4. The variety of datasets also applies to metrics. There are numerous evaluation metrics with a few stand out measurements. There is a difference between evaluating the generated images produced by a GAN and the model itself. In [41] Borji presents and discusses the pros and cons of evaluation measurements for GANs. The author creates a list with important characteristics an evaluation method should fulfil and tests evaluation methods against these characteristics. Any method used to evaluate GANs should favour high fidelity samples with high diversity which have disentangled latent spaces, well-defined boundaries, are sensitive to transformations and distortions, can withstand human judgment and have low complexity.

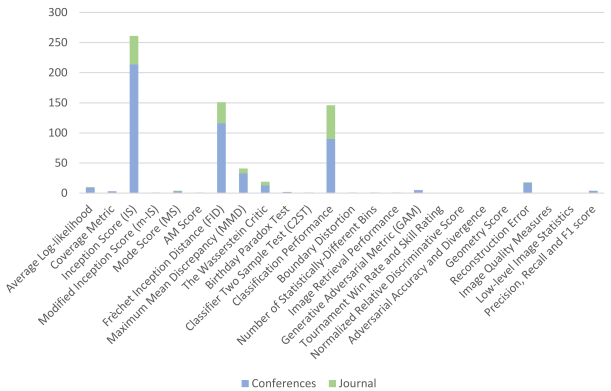

Figure 6. GAN evaluation metrics overview [41] and literature excerpt

24 quantitative measurements and 5 qualitative measurements are listed. The following Figure shows the quantitative metrics collected by [41] and shows a literature excerpt to give an indication of the metrics acceptance within the scientific community. Full text as well as abstract searches were

TABLE I. DATASET OVERVIEW OF COMMON GAN ARCHITECTURES

Architecture	MNIST	CIFAR-10	TFD	SVHN	FFHQ	ImageNet	MIR Flickr 25,000	YFCC100M 2	CelebA	LSUN
GAN	X	X	X							
cGAN	X					X	X	X		
InfoGAN	X			X					X	
WassersteinGAN										X
DCGAN	X	X		X		X			X	X
StyleGAN					X					X

TABLE III. DATASET OVERVIEW STACKGAN AND CP-GAN

Architecture	CIFAR-10	CLEVR	Flying Chairs	CUB	MS COCO	Oxford-102	LSUN	ImageNet
StackGAN	X	X	X					X
CP-GAN				X	X	X	X	

conducted for IEEE Xplore and ACM Digital Library with the concatenated search term:*"Full Text Only":"generative adversarial networ*" or "gan" and "metric name" and "Abstract":"generative adversarial networ*"*. Figure 6 shows the most used metrics for GAN evaluation namely Inception Score, Fréchet Inception Distance, Classification Performance and the Maximum Mean Discrepancy, summarized in the next sections.

C. Inception Score (IS)

The Incetion Score [42] applies the Inception Model (a pretrained model) to every generated image to receive $p(y|x)$ (conditional label distribution), ideally having a low entropy, meaning that the generated images have meaningful objects. The marginal $\int p(y|x = G(z))dz$ will have a high entropy if the model generated images of high variety. This results in the following metric:

$$exp(\mathbb{E}_x \mathrm{KL}(p(y|x)\|p(y))) \tag{22}$$

D. Fréchet Inception Distance (FID)

The Fréchet Inception Distance [43] is an improvement of the above described Inception Distance. The coding layer of the Inception Model is used to replace x by generalizing polynomials x (mean and covariance) to obtain vision related features. The coding units follow a multidimensional Gaussian. The distance between the two Gaussian (real-world and synthetic samples) the Fréchet distance (Wasserstein-2 distance) to assess the quality of generated samples given by the following equation:

$$d^2((m, C, (m_w, C_w))) = |\|m - m_w\|_2^2 + Tr\left(C + C_w - 2\left(CC_w\right)^{1/2}\right) \tag{23}$$

E. Classification Performance

Classification performance summarized in [41] is to apply unsupervised representation learning algorithms as feature extractors for labeled datasets and then evaluate their performance.

F. Maximum Mean Discrepancy (MMD)

The Maximum Mean Discrepancy [44][45] is a test to determine if the distribution of p and q are different based on drawn samples from each of them. This is done by finding a smooth function with the properties of being large if the points were drawn from p and small for samples drawn from q. The difference of values from a mean function between two samples is called the MMD. The greater the distance the more likely it is for the taken samples to come from different distributions. The goal of MMD is to answer if $p \neq q$ when p and q are distributions from x and $X := \{x_1, \ldots, x_m\}$, $Y := \{y_1, \ldots, y_n\}$ are distributed from p and q independently and identically.

VI. GAN Challenges

Application domains as well as specialized GAN architectures are flourishing and the potential and advances seem promising to not only achieve results of higher quality with better performance but also to stabilize the training process and prevent non-convergence or mode collapse. This section, will describe two occurring challenges in accordance with RQ5 for GANs as well as recent advances and new insights.

A. Mode Collapse and Non-Convergence

GAN networks are still prone to some problems like mode collapse or non-convergence. Mode collapse [46][47] is lack of diversity in generated samples. The problem lies in the minimax game where G tries to fool D. If G finds a sweet-spot (i.e., concentrate on a single mode), G is more likely to abuse this sweet-spot to produce a more plausible output. D is able to learn the pattern of this sweet-spot and can always classify it to be fake. The next iteration of D can get stuck and G can abuse this by creating the best output for the current D.

Non-convergence [46][48] is D's problem to distinguish between real and fake samples. With G improving, D's performance will get worse. With a near perfect G, D is forced to randomize its output (coin flip) and the feedback becomes negligible. The problem is if G is trained past this point and adjust to given feedback, the quality of generated samples will drop. Depending on the architecture and dataset some GANs are more prone to non-convergence as others.

B. Advances in Research

Improved architectures enabling higher resolutions, stabilizing training preventing mode collapse or non-convergences will further strengthen the applicability of GANs. This subsection will introduce some advances in research compliant with RQ6. In [49] Chavdarova, Gidel, Fleuret and Lacoste-Julien, show that training a GAN with stochastic gradient noise can prevent the convergence of standard game optimization methods. Quality aware GANs are proposed in [50]. The authors use a variation of structural similarity (SSIM) index and a quality aware discriminator gradient penalty function as regularizers for GANs. Learning disentangled representations unsupervised is proposed in [51]. By introducing a similarity constraint the authors show, that their approach is able to distinguish different representations by using conditions. In [52] a super resolution reconstruction method is introduced.

The combination of GANs and wavelet transformation used to train wavelet decomposition coefficients improves the quality of reconstructed images. Adiga, Attia, Chang and Tandon [53] propose two performance metrics mode-collapse divergence (MCD) and Generative Quality Score (GQS) to measure the quality of generated samples, created to capture the impact of mode collapse. Yang, Li, Qi and Lyu [54] introduce a method to predict images synthesized by GANs. The authors identified a difference between the location of facial landmarks due to lacking global constraints. Another approach by McCloskey and Albright [55] show that normalization steps by G leads to a detectable suppression of image characteristics.

VII. Conclusion

The presented survey summarizes the evolution of GANs by providing an overview of conducted research of the last 5 years, showing different architectural designs and loss functions. GANs will continue to evolve and further reshape deep generative models to produce more realistic samples. Image and video augmentation, restoration or 3D modelling are only a few domains in which GANs are already helping to produce new samples. The research contributing to GANs will find approaches to further stabilize training, minimize mode collapse and non-convergence. The varying architectural patterns use different datasets and metrics for evaluation. There is still a need to find a uniform accepted evaluation method consisting of datasets and metric. As technology advances, so will GANs and their possibilities. New generator and discriminator architectures will help to achieve the ambitious goal of generating realistic samples. This will be one of the biggest challenges as shown in Section VI-B as synthesized samples are still distinguishable from real samples. There are various domains (e.g., autonomous driving vehicles) with occurrences of rare high impact events, that are hard to capture (e.g., pedestrian crossing the street in close proximity to the vehicle). Combining the various fields of open challenges will help generating indistinguishable fake samples of such rare events, required for the evaluation of existing machine learning models to ensure they are able to recognize these rare events.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680, [Online] http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf [retrieved: 05-2020].

[2] M. Brückner and T. Scheffer, "Nash equilibria of static prediction games," in Advances in Neural Information Processing Systems 22, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 171–179, [Online] http://papers.nips.cc/paper/3755-nash-equilibria-of-static-prediction-games.pdf [retrieved: 05-2020].

[3] X. Chen and X. Deng, "Settling the complexity of two-player nash equilibrium," in 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), Oct 2006, pp. 261–272.

[4] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," nov 2014, [Online] http://arxiv.org/abs/1411.1784 [retrieved: 05-2020].

[5] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2019-June. IEEE Computer Society, jun 2019, pp. 4396–4405.

[6] W. Fedus, I. Goodfellow, A. M. Dai, and G. Brain, "MaskGAN: Better Text Generation via Filling in the," Tech. Rep., feb 2018.

[7] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in Proceedings of the IEEE International Conference on Computer Vision, vol. 2017-Octob. Institute of Electrical and Electronics Engineers Inc., dec 2017, pp. 2242–2251.

[8] Y. Chen, Y. Pan, T. Yao, X. Tian, and T. Mei, "Mocycle-GAN: Unpaired video-to-video translation," in MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia. New York, NY, USA: Association for Computing Machinery, Inc, oct 2019, pp. 647–655, [Online] http://dl.acm.org/doi/10.1145/3343031.3350937 [retrieved: 05-2020].

[9] Y. Zhang, X. Zhao, M. Kampffmeyer, and M. Tan, "DTR-GAN: Dilated temporal relational adversarial network for video summarization," in ACM International Conference Proceeding Series. New York, New York, USA: Association for Computing Machinery, may 2019, pp. 1–6, [Online] http://dl.acm.org/citation.cfm?doid=3321408.3322622 [retrieved: 05-2020].

[10] R. Arandjelović and A. Zisserman, "Object Discovery with a Copy-Pasting GAN," may 2019, [Online] http://arxiv.org/abs/1905.11369 [retrieved: 05-2020].

[11] R. Hermoza and I. Sipiran, "3D reconstruction of incomplete archaeological objects using a generative adversarial network," in ACM International Conference Proceeding Series. New York, New York, USA: Association for Computing Machinery, jun 2018, pp. 5–11. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3208159.3208173

[12] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 8, aug 2019, pp. 1947–1962.

[13] M. Lucic, K. Kurach, M. Michalski, O. Bousquet, and S. Gelly, "Are Gans created equal? A large-scale study," in Advances in Neural Information Processing Systems, vol. 2018-Decem, 2018, pp. 700–709.

[14] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in Advances in Neural Information Processing Systems, 2016, pp. 2180–2188, [Online] https://arxiv.org/abs/1606.03657. [retrieved: 05-2020].

[15] S. Mohamed and D. Jimenez Rezende, "Variational information maximisation for intrinsically motivated reinforcement learning," in Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2125–2133, [Online] http://papers.nips.cc/paper/5668-variational-information-maximisation-for-intrinsically-motivated-reinforcement-learning.pdf [retrieved: 05-2020].

[16] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," jan 2017, [Online] http://arxiv.org/abs/1701.07875 [retrieved: 05-2020].

[17] J. Gao and H. Tembine, "Distributionally Robust Games: Wasserstein Metric," in Proceedings of the International Joint Conference on Neural Networks, vol. 2018-July. Institute of Electrical and Electronics Engineers Inc., oct 2018.

[18] J. Heinonen, "Lectures on Analysis on Metric Spaces." Springer, New York, NY, 2001, ch. 6 - Lipschitz Functions, pp. 43–48.

[19] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2016, [Online] http://arxiv.org/abs/1511.06434 [retrieved: 05-2020].

[20] M. Mehralian and B. Karasfi, "RDCGAN: Unsupervised representation learning with regularized deep convolutional generative adversarial networks," in 2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium, AIAR 2018. Institute of Electrical and Electronics Engineers Inc., dec 2018, pp. 31–38.

[21] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 694–711.

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5967–5976.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-793-1

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[24] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in International Conference on Learning Representations, 2018, [Online] https://openreview.net/forum?id=Hk99zCeAb [retrieved: 05-2020].

[25] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks," in Proceedings of the IEEE International Conference on Computer Vision, vol. 2017-Octob. Institute of Electrical and Electronics Engineers Inc., dec 2017, pp. 5908–5916.

[26] S. Y. Choi, H. J. Jeong, K. S. Park, and Y. G. Ha, "Efficient Driving Scene Image Creation Using Deep Neural Network," in 2019 IEEE International Conference on Big Data and Smart Computing, BigComp 2019 - Proceedings. IEEE, feb 2019, pp. 1–4, [Online] https://ieeexplore.ieee.org/document/8679269/ [retrieved: 05-2020].

[27] A. Vobecký, M. Uřičář, D. Hurych, and R. Škoviera, "Advanced Pedestrian Dataset Augmentation for Autonomous Driving," Tech. Rep., 2019.

[28] M. Uřičář, P. Křížek, D. Hurych, I. Sobh, S. Yogamani, and P. Denny, "Yes, we GAN: Applying adversarial techniques for autonomous driving," Electronic Imaging, no. 15, feb 2019, pp. 48–1–48–17, [Online] http://arxiv.org/abs/1902.03442 [retrieved: 05-2020].

[29] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, "Deeproad: GaN-based metamorphic testing and input validation framework for autonomous driving systems," in ASE 2018 - Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. Association for Computing Machinery, Inc, sep 2018, pp. 132–142.

[30] V. V. Kniaz, F. Remondino, and V. A. Knyaz, "Generative Adversarial Networks for Single Photo 3D Reconstruction," in ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 42, no. 2/W9. Copernicus GmbH, jan 2019, pp. 403–408, [Online] https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W9/403/2019/ [retrieved: 05-2020].

[31] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in 2015 IEEE International Conference on Computer Vision (ICCV), Dec 2015, pp. 2758–2766.

[32] Y. Nakahira and K. Kawamoto, "DCVGAN: Depth Conditional Video Generation," in Proceedings - International Conference on Image Processing, ICIP, vol. 2019-Septe. IEEE Computer Society, sep 2019, pp. 749–753.

[33] S. Tulyakov, M. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2018, pp. 1526–1535.

[34] L. von Werra, M. Schöngens, E. D. Gamsiz Uzun, and C. Eickhoff, "Generative adversarial networks in precision oncology," in Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ser. ICTIR '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 145–148, [Online] https://doi.org/10.1145/3341981.3344238 [retrieved: 05-2020].

[35] S. Xie, H. Xu, and H. Li, "Artifact removal using gan network for limited-angle ct reconstruction," in 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA), Nov 2019, pp. 1–4.

[36] Y. Wen, R. Singh, and B. Raj, "Face Reconstruction from Voice using Generative Adversarial Networks," Tech. Rep., 2019, [Online] https://github.com/cmu-mlsp/reconstructing_faces_from_voices [retrieved: 05-2020].

[37] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," in International Conference on Learning Representations, 2019, [Online] https://openreview.net/forum?id=H1xQVn09FX [retrieved: 05-2020].

[38] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," Journal of Big Data, vol. 6, no. 1, dec 2019.

[39] Y. Wang, W. Zhang, and P. Chen, "Chinastyle: A mask-aware generative adversarial network for Chinese traditional image translation," in SIGGRAPH Asia 2019 Technical Briefs, SA 2019. New York, New York, USA: Association for Computing Machinery, Inc, nov, pp. 5–8, [Online] http://dl.acm.org/citation.cfm?doid=3355088.3365148 [retrieved: 05-2020].

[40] Y. Ito, W. Shimoda, and K. Yanai, "Food image generation using a large amount of food images with conditional gan: Ramengan and recipegan." New York, NY, USA: Association for Computing Machinery, 2018, [Online] https://doi.org/10.1145/3230519.3230598 [retrieved: 05-2020].

[41] A. Borji, "Pros and cons of GAN evaluation measures," Computer Vision and Image Understanding, vol. 179, feb 2019, pp. 41–65.

[42] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in Advances in Neural Information Processing Systems, 2016, pp. 2234–2242.

[43] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," Tech. Rep., 2017.

[44] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A Kernel Method for the Two-Sample-Problem," Tech. Rep., [Online] www.kyb.mpg.de/bs/people/arthur/mmd.htm [retrieved: 05-2020].

[45] I. Tolstikhin, B. K. Sriperumbudur, and B. Schölkopf, "Minimax estimation of maximum mean discrepancy with radial kernels," in Advances in Neural Information Processing Systems, 2016, pp. 1938–1946.

[46] Z. Zhang, M. Li, and J. Yu, "On the convergence and mode collapse of GAN," in SIGGRAPH Asia 2018 Technical Briefs, SA 2018. New York, New York, USA: ACM Press, dec, pp. 1–4, [Online] http://dl.acm.org/citation.cfm?doid=3283254.3283282 [retrieved: 05-2020].

[47] "Common Problems - Generative Adversarial Networks - Google Developers," [Online] https://developers.google.com/machine-learning/gan/problems [retrieved: 05-2020].

[48] "GAN Training - Generative Adversarial Networks - Google Developers," [Online] https://developers.google.com/machine-learning/gan/training [retrieved: 05-2020].

[49] T. Chavdarova, G. Gidel, F. Fleuret, and S. Lacoste-Julien, "Reducing noise in gan training with variance reduced extragradient," in Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 393–403, [Online] http://papers.nips.cc/paper/8331-reducing-noise-in-gan-training-with-variance-reduced-extragradient.pdf [retrieved: 05-2020].

[50] K. Parimala and S. Channappayya, "Quality aware generative adversarial networks," in Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 2948–2958, [Online] http://papers.nips.cc/paper/8560-quality-aware-generative-adversarial-networks.pdf [retrieved: 05-2020].

[51] X. Li, L. Chen, L. Wang, P. Wu, and W. Tong, "SCGAN: Disentangled Representation Learning by Adding Similarity Constraint on Generative Adversarial Nets," IEEE Access, vol. 7, 2019, pp. 147 928–147 938.

[52] Z. Huang and C. Jing, "Super-resolution reconstruction method of remote sensing image based on multi-feature fusion," IEEE Access, vol. 8, 2020, pp. 18 764–18 771.

[53] S. Adiga, M. A. Attia, W. T. Chang, and 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 - Proceedings. Institute of Electrical and Electronics Engineers Inc., feb 2019, pp. 1184–1188.

[54] X. Yang, Y. Li, H. Qi, and S. Lyu, "Exposing GAN-synthesized Faces Using Landmark Locations," in Proceedings of the ACM Workshop on Information Hiding and Multimedia Security - IH&MMSec'19. New York, New York, USA: ACM Press, jul 2019, pp. 113–118, [Online] http://dl.acm.org/citation.cfm?doid=3335203.3335724 [retrieved: 05-2020].

[55] S. McCloskey and M. Albright, "Detecting GAN-Generated Imagery Using Saturation Cues," in Proceedings - International Conference on Image Processing, ICIP, vol. 2019-September. IEEE Computer Society, sep 2019, pp. 4584–4588.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-793-1