

Fundamentals and Challenges of Generative Adversarial Networks for Image-based Applications

^{1st} Vinicius Luis Trevisan de Souza
Universidade Federal do ABC
Santo André, Brazil
vinicius.trevisan@ufabc.edu.br

^{2nd} Bruno Augusto Dorta Marques
Universidade Federal do ABC
Santo André, Brazil
bruno.marques@ufabc.edu.br

^{3rd} João Paulo Gois
Universidade Federal do ABC
Santo André, Brazil
joao.gois@ufabc.edu.br

Abstract—Significant advances in image-based applications have been achieved in recent years, many of which are arguably due to recent developments in Generative Adversarial Networks (GANs). Although the continuous improvement in the architectures of GAN has significantly increased the quality of synthetic images, this is not without challenges such as training stability and convergence issues, to name a few. In this work, we present the fundamentals and notable architectures of GANs, especially for image-based applications. We also discuss relevant issues such as training problems, diversity generation, and quality assessment (metrics).

Index Terms—Generative Adversarial Network, image manipulation, deep image synthesis, deep neural network

I. INTRODUCTION

Generative Adversarial Networks (GANs) techniques have received much attention in the fields of image processing, computer vision, and computer graphics due to their appealing results. They are based on the so-called “adversarial training”, which consists of two components: a generator and a discriminator that compete with each other [1], [2]. In particular, the generator aims at synthesizing fake data making it realistic enough to fool the discriminator, which in turn tries to detect these fake instances.

Among the image-based tasks performed by GANs, we highlight super-resolution, style transfer, image manipulation and synthesis, and image-to-image translation (Fig. 1). However, it is worth noting that GANs are also used in other domains, such as sequential data synthesis, text, and audio [3].

II. IMAGE-BASED APPLICATIONS

GANs that are based on convolutional neural networks (CNNs) can learn the representation of the features of the images during their training, which allows the development of various applications. We mention here the transformation of images between different domains, the composition of images from collages of these features, style transfer, super-resolution and even photo restoration.

A. Image translation and style transfer

Image-to-image translation transforms an input image into an output image with different attributes. Methods such as Pix2Pix [5] or CycleGAN [6] can translate semantic maps into landscapes, labels into facades, horses into zebras, and even day into night. Style transfer is another related application

where images retain their shape but receive style (mainly color and textures) from another source. Examples include converting photographs into paintings [6] or changing the texture of a building [7].

B. Image synthesis

Methods such as GauGAN [4] and Pix2PixHD [8] can reconstruct scenes and landscapes from an input in the form of a semantic map. Their goal is to generate realistic images based only on the latent representation learned from the generator. Other methods focus on creating synthetic faces [9], [10] even in different poses [11]. It has been shown that image synthesis methods based on GANs work very well for specific classes. However, generating images from different classes with a single network is still a challenge [3], [12].

C. Image manipulation

In certain cases where the latent space mapped by the network is disentangled [10], it is possible to manipulate the properties of the synthetic images by performing operations on their latent vectors. Karras et al. [10] demonstrate this property by independently modifying features such as gender and age of faces created with the StyleGAN network. To recover the latent vector that would produce a given image in an already trained network, and thus manipulate it, one must often use GAN inversion techniques [13]–[15].

D. Super resolution and image repair

Super-resolution approaches estimate a high-resolution image from a low-resolution input [16]. Specialized super-resolution GANs can greatly increase the spatial resolution of images while preserving small details [16]–[18]. Similar techniques can be used to restore old photographs [19] or to complete missing parts of images [20].

III. FUNDAMENTALS

A. GAN architecture and adversarial training

The Generative Adversarial Networks proposed by Goodfellow *et al.* [1] consists of two main components: a discriminator D trained to distinguish between true and false input data, and a generator G that generates synthetic (false) samples similar to the true ones from a noise vector input $z \in Z$, where Z is the latent space of the GAN, usually derived from a probability

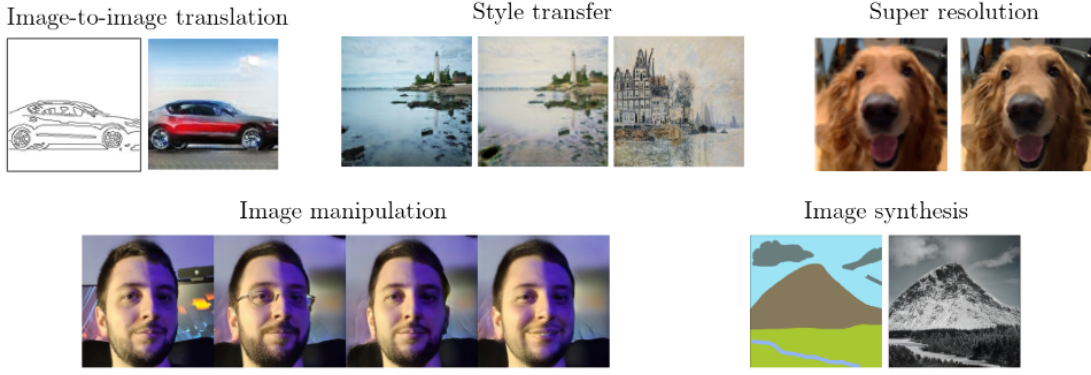


Fig. 1. Image-based GANs applications. In “image-to-image translation,” the first image is a sketch and the second is the reconstruction of a car based on it. In “style transfer,” the first image is the content, the last is the style, and the middle is the result of applying the style to the original picture. In “super resolution,” the first image with dimensions 64×64 , was scaled up to 256×256 (second image). In “image manipulation,” the first image is the input, while subsequent images are manipulations of age, pose, and smile. In “image synthesis”, NVIDIA Canvas was used to create the image of a landscape from a semantic map using GauGAN [4].

distribution. Typically, the discriminator and the generator are built with artificial neural networks.

Let p_{data} be a real data distribution and p_z be the distribution from which the input vectors are sampled. The generator G , trained with parameters θ_z , acts as $G(z, \theta_z) : p_z \rightarrow p_g$, where p_g is the synthetic data distribution generated by G . The goal of the GAN is to have $p_g \sim p_{data}$ after training. For each iteration, the discriminator D receives as input both a sample of real data x_{real} and a synthetic example created by the generator $G(z)$.

The discriminator is correct whenever it classifies the real input as true and the synthetic input as false. If the discriminator is wrong, it corrects itself to perform better on the next iteration. In turn, the generator is penalized if the discriminator classifies $G(z)$ as false (Fig. 2).

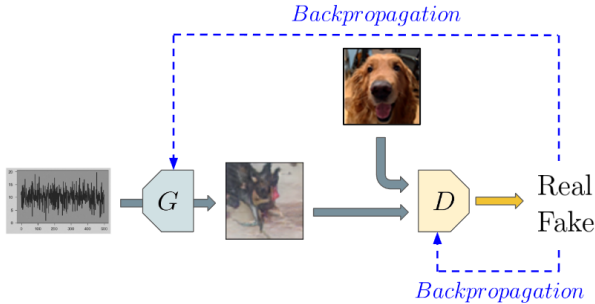


Fig. 2. Training an image GAN. The generator produces a synthetic (fake) image from a noise input. The discriminator is applied to synthetic and original images and classifies them as real (1) or fake (0). The loss is evaluated and then the parameters of both networks are updated by backpropagation.

B. Adversarial loss

The generator and discriminator participate in a min-max training represented by Eq. 2 [1]. Interpreting $D(x)$ as the probability that the data x come from the distribution p_{data} rather than p_g , the discriminator is trained to maximize the probability of correctly discriminating the real images as true

(1) and discriminating the fake ones as false (0). In turn, the generator is trained to produce synthetic images that the discriminator assigns a high probability of being true. Given that G^* and D^* represent the trained generator and discriminator, respectively, the objective of the training is:

$$G^*, D^* = \min_G \max_D \mathcal{L}(D, G), \quad (1)$$

with

$$\begin{aligned} \mathcal{L}(D, G) = & \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] \\ & + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))], \end{aligned} \quad (2)$$

where the first term of Eq. 2 is the expected probability of the discriminator D to correctly classify the real input x as taken from the distribution p_{data} , whereas the second term is the expected probability of the discriminator correctly classify the synthetic input $G(z)$ as taken from the distribution p_g , and consequently that z is from the distribution p_z . Equation 2 is called *GAN loss* or *adversary loss*.

C. Deep Convolutional GANs

The *Deep Convolutional GANs* (DCGANs) [21] extended the original GANs by using convolutional networks in their architecture. In the generator, a noise vector input is processed through layers of transposed convolutions that sequentially synthesize the output image. The discriminator then receives real and fake images and uses the convolutions to capture the image attribute representation and guide the training of the generator with the adversarial training strategy.

The authors [21] have shown that this architecture can synthesize realistic images and that the trained discriminator can also be used in image classification tasks, with competitive performance compared to previous supervised methods.

D. Conditional GANs

An issue of the original GAN and DCGAN is that the composition of the generated image depends almost entirely

on the information contained in the noise vector, so there is little control over what the generator will create.

Conditional GANs (CGANs) [22] contain an additional vector y that conditions the generation of synthetic data. This vector may contain the class to which the data belongs, or some other type of condition, such as natural language annotations, or even image inputs. With the addition of y , the *loss* of the CGAN is

$$\mathcal{L}_{CGAN}(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z|y)))] \quad (3)$$

while the generator and discriminator aim to optimize:

$$G^*, D^* = \min_G \max_D \mathcal{L}_{CGAN}(D, G). \quad (4)$$

In the generator, conditioning can be done by concatenating the noise vector z with the condition vector y . Similarly, the discriminator would obtain the concatenation of the data x with the same condition vector y .

With CGAN, it was possible to select which characters of the MNIST dataset would be generated instead of creating a random character.

IV. REMARKABLE ARCHITECTURES

A. Pix2Pix

Pix2Pix [5] is a conditional GAN, designed to perform supervised image transformations between different domains. For example, it has been used in translating semantic annotations into street photos with cars, annotations into facades, grayscale into colored images, aerial pictures into maps, day images into night images, and edges into photos.

This architecture is built after a conditional GAN (CGAN [22]), where both the generator and discriminator receive an image x_A as a “condition”. However, the generator does not receive the noise input that is present in the original CGAN (Fig. 3). Training is performed on image pairs (x_A, x_B) , where x_A is the input image and x_B is the target to be constructed by the generator.

Pix2Pix also uses CGAN’s *loss* (Eq. 3) for both the generator and the discriminator, but regularization is added so that the synthetic image $G(x_A)$ contains the features of the objective x_B . This regularization is the pixel-to-pixel L1 distance between the synthetic image and the target. The generator is trained to minimize \mathcal{L}_G while the discriminator seeks to maximize \mathcal{L}_D :

$$\mathcal{L}_G = \mathcal{L}_{CGAN}(G, D) + \lambda \|x_B - G(x_A)\|_1, \quad (5)$$

$$\mathcal{L}_D = \mathcal{L}_{CGAN}(G, D), \quad (6)$$

where $\mathcal{L}_{CGAN}(G, D)$ is the CGAN *loss* and $\|x_B - G(x_A)\|_1$ is the L1 norm (Manhattan distance) of the images. The parameter λ controls the effect of the L1 term on the training of the generator.

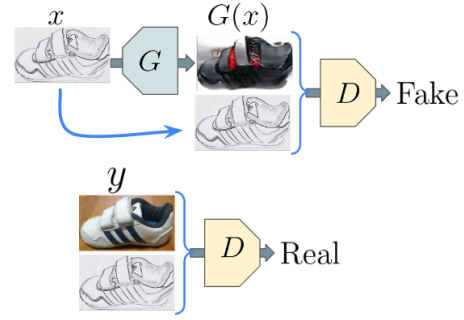


Fig. 3. Pix2Pix simplified architecture. On the top, a sketch input x is presented to the generator G , which produces a synthetic image $G(x)$. The discriminator D receives a set containing this image as well as the original image x and should classify this set as “fake”. On the bottom, the discriminator is presented with a real image y along with the sketch input x , and in this case the discriminator should classify the set as “real”. With an ideal generator, the image y and the synthetic image $y^* = G^*(x)$ should be identical.

B. CycleGAN

CycleGAN [6] proposes an unsupervised approach to image-to-image translation between two different domains based on cycle-consistent GANs, i.e., consistently transforming images from one domain to the other and returning to the original domain with only a small reconstruction error. Since CycleGAN is unsupervised, there is no need to use paired databases like those used by Pix2Pix.

Let A and B be two different domains, but with comparable properties, such as horses and zebras, or landscape pictures and paintings. A generator $G_A : A \rightarrow B$ performs the transformation $x_B = G_A(x_A)$, which transforms the image x_A from domain A to a synthetic image x_B corresponding to its equivalent in domain B . Similarly, a generator $G_B : B \rightarrow A$ transforms an image y_B in domain B into the corresponding image $y_A = G_B(y_B)$ in domain A .

To ensure that the transformation $A \rightarrow B$ proceeds correctly, there is a discriminator D_B that classifies images as belonging or not belonging to B , trained with real images of B and images synthesized from inputs of A . There is also a discriminator D_A that similarly evaluates whether images belong to A or not. A sketch of the architecture can be seen in Figure 4.

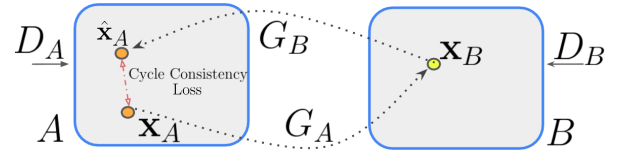


Fig. 4. CycleGAN Architecture. The generator G_A transforms an image from domain A to domain B , while G_B does the reverse. D_A is a discriminator that evaluates whether the images belong to domain A and D_B does the same for domain B . By inputting an image x_A sequentially through G_A and G_B gives \hat{x}_A , and the distance from the original image is the cycle consistency loss.

However, cycle consistency can only be achieved if any image x_A can undergo two successive transformations, return-

ing to its original domain as $\hat{x}_A = G_B(G_A(x_A))$, and if the distance $|x_A - \hat{x}_A|$ is within a small permissible error range (Fig. 4). This must also be true for any image from domain B . The authors then proposed a cycle consistency loss:

$$\mathcal{L}_{cyc}(G_A, G_B) = \|G_B(G_A(x_A)) - x_A\|_1 + \|G_A(G_B(x_B)) - x_B\|_1. \quad (7)$$

With this new restriction, both generators are encouraged to make only small changes to the shape and focus on transforming only the most important attributes that distinguish the images of A from those in B . The complete system loss is defined by:

$$\begin{aligned} \mathcal{L}(G_A, G_B, D_A, D_B) = & \mathcal{L}_{CGAN}(G_A, D_B) \\ & + \mathcal{L}_{CGAN}(G_B, D_A) \\ & + \gamma \mathcal{L}_{cyc}(G_A, G_B), \end{aligned} \quad (8)$$

where \mathcal{L}_{CGAN} is the CGAN adversarial loss, and γ is used to control the impact of the cycle-consistency loss on the training.

Although the CycleGAN architecture has the advantage of being an unsupervised method, it is not capable of drastically changing the shape and structure of the depicted objects, but is mainly suitable for texture-based transformations.

C. ProGAN

Karras *et al.* [9] proposed a training strategy for GANs that can generate realistic images at high resolution (1024^2). In this method, both the generator and discriminator start their training with low-resolution images (4^2) and gradually increase the resolution by adding more layers. This progressive growth GAN is called ProGAN and is shown in Fig. 5.

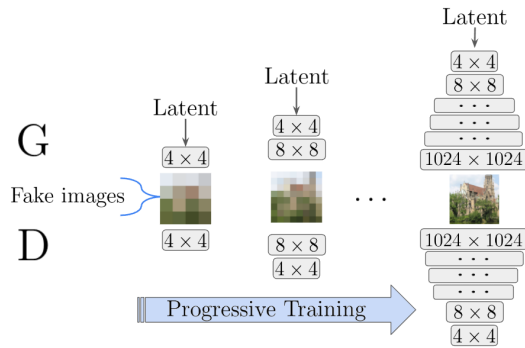


Fig. 5. ProGAN: progressive growth of GANs. The generator and discriminator networks first operate on low-resolution images (4×4) for a predefined number of epochs (iterations). Then another set of layers is added, and the network begins to work with (8×8) images. This method grows step by step until the network reaches the target resolution of 1024×1024 .

Other differences of this approach compared to other networks are the use of PixelNorm as the normalization layer and the use of a balanced learning rate, which ensures that the learning speed is the same for all weights. The loss used in this architecture is Wasserstein loss with Gradient Penalty (WGAN-GP) [23], [24].

The progressive training approach leads to a more stable training of GANs [9] and allowed the development of more advanced techniques with even better results [10], [25].

D. StyleGAN

The StyleGAN architecture is based on ProGAN, but with new additions to the generator [10]. One of the most important changes is the mapping network, which consists of eight Dense MLP layers that transform the input noise vector z into another vector w , effectively changing the shape of the latent space learned by the network.

As a consequence, the learned representation exhibits a high level of disentanglement, meaning that the image features in the latent space W [10] are linearly separable and the feature regions are well-defined so that they can be used in a binary setup (e.g., young vs. old). This allows manipulation of individual facial features by simple operations on the intermediate vector in the latent space W [10], [15].

In the generator architecture, the vector w is added to each resolution layer and carries its information to guide the creation of details at different scales. A fixed Gaussian noise array is also added to these layers so that the generator can produce stochastic details such as hair or freckles.

V. DISCUSSION TOPICS

A. Training challenges

A notorious fact regarding GANs is the difficulty to train them correctly, mainly due to convergence and stability issues [3], [26], [27].

One of the main problems is mode collapse, where the gradient of the discriminator tends to guide the generator in creating images from a single mode [26], resulting in very similar images with few variations. Another common issue is the vanishing gradient that occurs when the discriminator approaches its saturation [23], [27].

After Arjovsky *et al.* showed that one of the main causes of instability in GAN training is the use of the KL-divergence as the original loss [27], they proposed the use of the Earth Mover distance (EM), or Wasserstein metric, to compose a nonsaturating adversarial loss, and called the resulting network Wasserstein-GAN (WGAN), thereby significantly reducing the effects of mode collapse and vanishing gradient [23], [24]. Other methods also used various loss functions and regularization terms to address these issues [3], [12].

The progressive training technique also helps to increase stability by splitting the learning task, which should be performed simultaneously by the entire network, into a layered approach. In this approach, for each new layer, the representation of the lower resolutions should have already been learned by the previous layers, so they only need slight updates to refine this representation, effectively reducing the instability of the whole network.

B. Diversity generation

One of the most challenging tasks for GANs is to train a single network capable of generating images for different

classes. It implies that a specialized GAN trained to create human faces might not be able to generate images from other domains, like animals or houses. Usually, GANs trained to create images from different domains cannot do so with the same level of quality as a specialized GAN would do in its own domain [3].

Regardless, some interesting approaches, such as Self-Attention GAN (SAGAN) [28] and BigGAN [29], use different architectures to address this problem. SAGAN uses a self-attention mechanism to capture global image features, complementing the convolution's emphasis on local features and allowing the learning of more diverse structures. The BigGAN was built based on the SAGAN architecture, and adds techniques to stably increase the network size (number of channels or layers), consequently increasing the number of trainable parameters and allowing the network to learn even more representations.

C. Quality evaluation

A widely discussed problem in the image synthesis area is evaluating the perceived quality of the generated images [12]. The authors of Pix2Pix [5] and CycleGAN [6] strived to create natural and realistic images. To this end, they evaluate the visual quality of the synthetic images using human questioning through perceptual studies on crowdsourcing platforms.

On computational assessments, Radford *et al.* [26] took advantage of the satisfactory performance of the InceptionV3 network in classifying images of the 1000 different ImageNet classes [30] to propose the Inception Score (IS), which consists in feeding synthetic images into this network, obtaining the conditional probabilities for each label and evaluating a score based on them. Although this metric correlates well with human judgement [26], it can present good results even for networks in mode-collapsed generators [12], [31].

Based on the Inception Score, Heusel *et al.* [31] proposed the Fréchet Inception Distance (FID), which uses the InceptionV3 network in both the real and fake image sets, thus generating two probability distributions which are compared by using the Fréchet distance. Even though the original FID uses InceptionV3 as the basis for the evaluation, other networks can also be used [12], [32].

At the moment, FID is one of the most used metrics to compare the result of different GANs, but this comparison is not always fair. Within the same architecture, an improvement in the FID is correlated with an improvement in the perceived quality of the created images. Nonetheless, when comparing different architectures, a better numeric value might not necessarily indicate a better GAN. It is due to the fact that FID is very dependent on the ImageNet classes in its evaluation, usually affected by textures more than by shapes [32].

Other measurements, such as KID [33] or MS-SSIM [34], can also be used, but they all have downsides. Because of this, the common practice to compare different methods has been to use more than one metric to evaluate the quality of the generated images.

D. Computing capacity

A particular point for attention when working with GANs is the computational cost associated with their development and training. Since they are composed of at least two networks with thousands of parameters to be adjusted, training can take hours or days.

One example of how expensive this can be, during all the development of the StyleGAN3, one or more NVIDIA DGX-1 clusters were used, consuming the equivalent of 91.77 years of processing of a single Volta GPU, and approximately 225 MWh of electricity [11].

This computational scale can make it difficult or even impossible for smaller research groups to be able to compete and even reproduce results such as those. Regardless, there is a significant potential for creating more cost-effective GANs.

E. Additional discussion topics

Even with the many advances of GANs on image synthesis tasks, there is still room for improvement in the training stability and generation of diverse, multi-class images. Further advancement may still come from the application of GANs in specific situations, such as video processing [35] and the generation of images from text, in which denoising diffusion probabilistic models [36] such as DALL-E 2 [37] and Imagen [38] have shown impressive results.

A substantial impact of generating plausible synthetic images is the inappropriate usage in forgery applications, which can harm people and institutions or transmit false information. Further studies could concentrate on detecting false images to avoid those situations [12].

VI. CLOSING REMARKS

We close this work with the very recent observation by Xiao *et al.* [39] since it complements our discussion (Sec. V) and opens doors for reflections and future improvements and applications. The authors argued that there are three requirements pursued by deep generative methods: (a) generating high-quality samples, (b) fast sampling, and (c) generating diversity samples/mode coverage. However, the generative approaches had only partially dealt with these requirements, defining the *generative learning trilemma* [39]. Works such as Denoising Diffusion GAN [39] and StyleGAN-XL [40] aim to tackle all three requirements. We believe that, from now on, the most relevant research advances will use this trilemma as their starting point. Besides, they will focus on hybrid approaches, mixing GANs, VAEs [41], [42], and diffusion models [43], expanding the domain of the generative models applications.

ACKNOWLEDGMENT

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

REFERENCES

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
- [2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [3] Z. Wang, Q. She, and T. E. Ward, "Generative adversarial networks in computer vision: A survey and taxonomy," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.
- [4] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2337–2346.
- [5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [6] 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*, 2017, pp. 2223–2232.
- [7] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. Efros, and R. Zhang, "Swapping autoencoder for deep image manipulation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7198–7211, 2020.
- [8] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8798–8807.
- [9] 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.
- [10] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.
- [11] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," in *Proc. NeurIPS*, 2021.
- [12] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [13] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan: How to embed images into the stylegan latent space?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4432–4441.
- [14] —, "Image2stylegan++: How to edit the embedded images?" in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8296–8305.
- [15] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain gan inversion for real image editing," in *European conference on computer vision*. Springer, 2020, pp. 592–608.
- [16] 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.
- [17] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *The European Conference on Computer Vision Workshops (ECCVW)*, September 2018.
- [18] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-esrgan: Training real-world blind super-resolution with pure synthetic data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1905–1914.
- [19] X. Wang, Y. Li, H. Zhang, and Y. Shan, "Towards real-world blind face restoration with generative facial prior," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [20] Y. Li, S. Liu, J. Yang, and M.-H. Yang, "Generative face completion," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5892–5900.
- [21] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," in *International Conference on Learning Representations*, 2016.
- [22] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
- [23] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.
- [24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.
- [25] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8110–8119.
- [26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, pp. 2234–2242, 2016.
- [27] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *International Conference on Learning Representations*, 2017.
- [28] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.
- [29] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2019.
- [30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [31] 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," *Advances in neural information processing systems*, vol. 30, 2017.
- [32] T. Kynkäänniemi, T. Karras, M. Aittala, T. Aila, and J. Lehtinen, "The role of imagenet classes in fréchet inception distance," *arXiv preprint arXiv:2203.06026*, 2022.
- [33] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in *International Conference on Learning Representations*, 2018.
- [34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [35] I. Skorokhodov, S. Tulyakov, and M. Elhoseiny, "Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3626–3636.
- [36] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 17–23 Jul 2022, pp. 16 784–16 804.
- [37] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.
- [38] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *arXiv preprint arXiv:2205.11487*, 2022.
- [39] Z. Xiao, K. Kreis, and A. Vahdat, "Tackling the Generative Learning Trilemma with Denoising Diffusion GANs," in *International Conference on Learning Representations (ICLR)*, 2022.
- [40] A. Sauer, K. Schwarz, and A. Geiger, "Stylegan-xl: Scaling stylegan to large diverse datasets," in *ACM SIGGRAPH 2022 Conference Proceedings*, ser. SIGGRAPH '22. New York, NY, USA: Association for Computing Machinery, 2022.
- [41] R. Wei, C. Garcia, A. El-Sayed, V. Peterson, and A. Mahmood, "Variations in variational autoencoders-a comparative evaluation," *Ieee Access*, vol. 8, pp. 153 651–153 670, 2020.
- [42] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 2018.
- [43] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.