A Review on Generative Adversarial Networks Architectures

Tahere Hemmati
Department of Computer Engineering
Faculty of Engineering
University of Guilan

Amir Abbasi
Department of Computer Engineering
Faculty of Engineering
University of Guilan

Sepideh Bahrami
Department of Computer Engineering
Faculty of Engineering
University of Guilan

***Abstract*—In recent years, generative adversarial learning has reached remarkable achievements in lots of research areas. GAN models are widely used in the computer vision field and have made significant advances in critical challenges like image-to-image translation, image generation, and similar domains. Despite remarkable achievements, few comprehensive reviews introduce and categorize different types of GAN models in terms of their architectures and applications. Hence, in this paper, we provide a comprehensive review of GAN models from the perspectives of mathematical theory, architectures, and their fine-grained training details. First, the mathematical background of GAN is presented in detail. Then, a broad type of GAN models is introduced from the architecture-variant perspectives. Furthermore, their configurations and advantages are pointed out. Finally, the present study summarizes the open research challenges, provides a valuable and comprehensive survey on GAN, and discusses future research directions and challenges.**

I. Introduction

Generative adversarial networks have become a topic of much researches. This idea was proposed by [1] which was inspired by min-max two-player games. GAN has become a milestone in deep learning, and despite traditional generation models, GAN can optimize complicated loss functions and deal with some potential challenges. A GAN model can learn complex and sharp distributions, generating high-quality images, and ignore biases. GAN is used in various domains, including computer vision, natural language processing, and series synthesis. GAN has achieved lots of successes, especially in computer vision, such as image generation, image super-resolution, image-to-image translation, image completion, and data augmentation.

Considering the stated advantages, it worth that assess some common problems of GAN models. It is expected that generative models be able to generate almost all types of images of distribution. However, in some cases the model attempts to cheat by learning a limited number of samples and minimizes its loss function using this trick. This problem is called *mode collapse* and several successful studies are performed to solve this problem [2]-[4]. Another common problem of GAN models is unstable training processes which leads to oscillating model parameters and it never converges.

In addition, evaluating GAN models is difficult and proper metrics are needed to measure similarity between the real and generated distribution. Unfortunately estimation of generated distribution is a big problem a makes this measuring a potential challenge. Hence, some successful researched were performed and proposed proper evaluation metrics [5]-[13].

Most of researches on GAN models an be considered in two main subjects: 1) proposing better methods for training and 2) employing GAN models in a wide range of applications. [14] discusses different types of GAN models on four benchmark datasets. Their outcomes confirmed that using original GAN model with spectral normalization provides a good performance when applying GAN models to a new dataset. Nevertheless, it should be noted that benchmark datasets do not consider diversity significantly and so their results only focus on assessing of image quality. [15] presents various type of GAN architectures and also different useful metrics used for evaluation. Their work did not compare performance of most of architectures, complexity, and their applications. [16]-[18] reviewed the newest research directions and applications of GAN and compared them through various applications.

Considering the most of reviews on GAN, we present a comprehensive survey on GAN from the perspectives of applications and wide type of architectures. We provide background concepts of generative adversarial learning and foundation of GAN models in section II. Then, we introduce various useful architectures and their applications in section III. In section IV, we briefly illustrate differences and relationships of introduced GAN architectures and also suggest some future research ideas. Finally, in section V, we conclude our study and highlight open research areas of GAN.

II. Generative Adversarial Networks

The minmax game theory inspires GANs. The main idea is to implement two opponent networks that try to compete and defeat each other. A simple type of GAN model is shown in Fig. 1. Generator attempts to fool discriminator and learn the distribution of real data to generate new similar samples. On the other hand, the discriminator gradually learns to distinguish between real and generated images. As can be seen, there is a competition between discriminator and generator, and these two networks should constantly increase their capability to win. In fact, better discriminator forces generator to learn better images. Loss function of GAN is defined as follows:

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

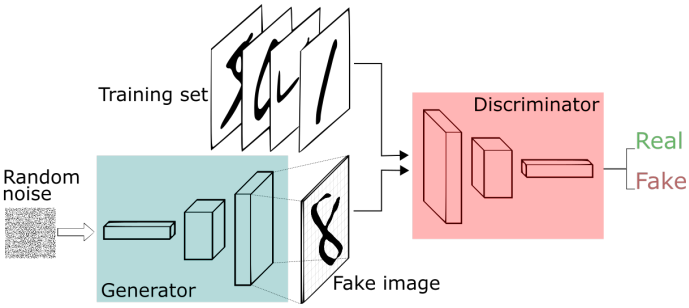

Fig. 1. Caption

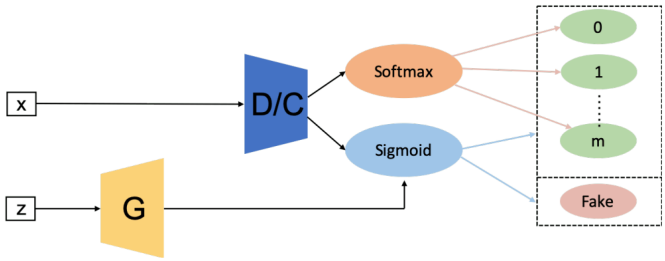

Fig. 2. SGAN Architecture (from [22])

where p_r is actual distribution, p_z is generated distribution, D is discriminator, and G is generator. This equation comprises loss of both generator and discriminator networks. First part of this equation is for optimizing the generator and the second part is the loss of discriminator. The aim is to find a Nash equilibrium between the two parts and improving the generator so that it learn distribution of data. According to this equation, real images entering discriminator are tagged as One and generated images are tagged as Zero and the goal of discriminator is to classify data sources correctly: true(for real images) and false (for fake images). On the other side, generator attempts to make its generated images similar to real images for discriminator and fool it. Generative ability of generator is improved until discriminator fails to specify fake images and in this case generator has learned the distribution of actual data.

III. GAN Architectures

A. Fully-connected GAN (FCGAN)

This architecture consists of fully connected neural networks in the generator and discriminator. It was initially practiced by the original GAN paper and was conducted for simple tasks such as MNIST [19], CIFAR [20], and Toronto Face dataset. This paper used Maxout [21] in the discriminator and a mixture of ReLU and sigmoid activations in the generator as the architecture choices. The designers propose to take k steps to optimize D and one step to optimize G due to the overfitting of the discriminator when the optimization of D ends in the internal learning loop. However, this GAN does not demonstrate good generalization performance for more complex image types.

B. Semi-supervised GAN (SGAN)

SGAN is motivated by semi-supervised learning [22]. Semi-supervised learning assigns to a learning problem (and algorithms composed for the learning problem) that includes a limited amount of labeled samples and many unlabeled examples from which a model must learn and make predictions on new examples. Opposed to FCGAN, the discriminator applied in SGAN is multi-headed, comprising softmax to classify the actual data and sigmoid activation functions to distinguish real and fake examples. Although this GAN accomplished some enhancements and better results on both its discriminator and generator, its architecture is still relatively simple, which affects the functionality of the model. Moreover, it was only trained on the MNIST dataset, limiting the domain choices for the succeeding real-world practice.

C. Conditional GAN (CGAN)

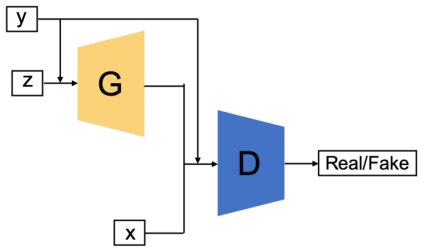

Fig. 3. CGAN Architecture (from[23])

The conditional generative adversarial network (CGAN) suggests the conditional generation of images by a generator model [23]. Image generation can be conditional upon the desired class label, letting the targeted produced images of an assigned kind. This approach is provided by feeding class labels or other modal data to both discriminator and generator models. As shown in Figure 3, Y is fed to discriminator and generator; however, it should be pointed out that Y is one hot encoded and concatenated with the input (x and z) to both models. CGAN strategy has successfully intensified the discriminative capability of the discriminator.

As represented in Equation 2, Since x and y are conditioned by z, this GAN's loss function is slightly different from the FCGAN (see Equation 1). Availing from the extra encoded y data, CGAN has brought GANs over the multimodal data generation. Hence, CGAN can work with unimodal image datasets and multimodal datasets such as Flickr containing labeled image data with their related user-generated metadata (UGM).

$$\min_G \max_D E_{x\sim p_r} log[D(x|y)] + E_{z\sim p_z} log[1 - D(G(z|y))] \tag{2}$$

The original paper demonstrated the CGAN experiments on the MNIST and Yahoo Flickr Creative Common 100M (YFCC

100M). For the MNIST, the model was trained using SGD with a mini-batch size of 100 and an initial learning rate of 0.1. Dropout was utilized with a probability of 0.5 to both generator and discriminator. The momentum was used with an initial value of 0.5 and finally was increased up to 0.7. Class labels were encoded as one-hot vectors and fed to both G and D. About YFCC 100M practice, they initiated with the same training hyperparameters as the settings in the MNIST experiment. While the encoded label's approach enhances the discriminator performance of CGAN, some of the generated labels still miss a connection with images.

D. InfoGAN

Alongside the noise vector z input, InfoGAN [24] provides control variables called the latent code (c) for its generator, targeting the significant structured semantic traits of the actual data distribution. InfoGAN tries to solve:

$$\min_G \max_D V_1(D,G) = V(D,G) - \lambda I(C; G(z,c)), \lambda > 0 \quad (3)$$

where $V(D,G)$ is the objective function of the original GAN, $G(z,c)$ is the generated example, I is the mutual information, and λ is a regularization parameter that can be tuned. By maximizing $I(c; G(z,c))$, the mutual information between c and G(z; c) will be maximized to construct c carry as many relevant and notable features of the authentic samples as potential. However, $I(c, G(z; c))$ is challenging to optimize directly in practice since it requires access to the posterior probability $P(c|x)$. To mitigate the challenge, we can have a lower bound of $I(c; G(z; c))$ by defining an auxiliary distribution $Q(c|x)$ to approximate $P(c|x)$. This approximation is achieved by training the generator via mutual information through a new model, called Q or the auxiliary model. The new model shares the same weights as the discriminator model for interpreting an input image except for the last layer. While the discriminator model predicts whether the image is real or fake, the auxiliary model predicts the control codes used to produce the image. This decision leads to the final InfoGAN equation:

$$\min_G \max_D V_1(D,G) = V(D,G) - \lambda L_I(c; Q), \lambda > 0 \quad (4)$$

where $L_I(c; Q)$ is the lower bound of $I(c; G(z; c))$. InfoGAN has experimented on MNIST, 3D face images [25], 3D chair images [26], SVHN [27], and CelebA datasets. All datasets share the same training configurations, in which batch normalization is applied, and Adam is used as the optimizer. Leaky ReLU with a rate of 0.1 is applied to the discriminator, and the ReLU is utilized for the generator. InfoGAN has several alternatives, such as causal InfoGAN [30] and semi-supervised InfoGAN (ss-InfoGAN) [31].

E. Auxiliary Classifier GAN (AC-GAN)

AC-GAN [28] is built on the idea of class conditional GANS (CGANS), which has a class-conditional prior input and

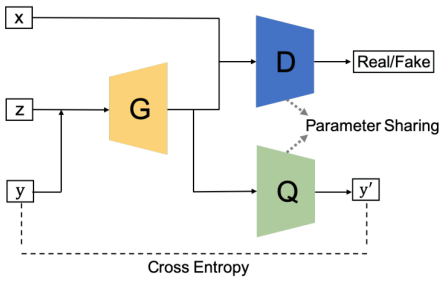

Fig. 4. InfoGAN Architecture (from [24])

challenges the discriminator with reproducing based on that class. Each generated example has a corresponding class label c in addition to z. The discriminator has to predict the real or fake and label the image corresponding to its actual class; therefore, it consists of a discriminator D (distinguishes real and fake samples) and a classifier Q (classifies real and fake samples). Similar to InfoGAN, the discriminator and classifier share all weights except the last layer. The loss function of AC-GAN can be formed regarding the discriminator and classifier, which can be stated as:

$$L_s = E_{x\sim p_r} log[D(x|c)] + E_{z\sim p_z} log[1 - D(G(z|c))] \quad (5)$$

$$L_c = E_{x\sim p_r} log[Q(x|c)] + E_{z\sim p_z} log[Q(G(z|c))] \quad (6)$$

where D is trained by maximizing $LS+LC$ and G is trained on maximizing $LC - LS$. In the original paper, AC-GAN was trained on the CIFAR-10 [29] and the ImageNet for all 1000 classes. For both datasets, ACGAN was trained with a 100 mini-batch size and Adam as the optimizer in D, G, and Q. The authors have experimented with ImageNet to gain the most competent number of classes in a model concerning the outcomes. Finally, they discovered that each model could perform its stablest when the number of classes is assigned to 10 as they experienced the mode collapse in the generator by increasing this number.

AC-GAN has improved visual quality for the generated images and has high model diversity. Furthermore, it can be extended to any generative framework. However, these developments rely on a large-scale labeled dataset, which may encounter some challenges in real-world demands.

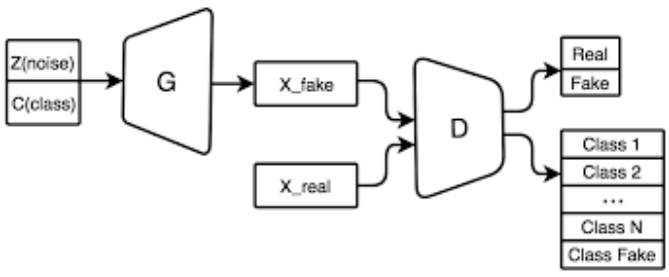

Fig. 5. ACGAN Architecture (from [])

F. Deep Convolutional GAN (DCGAN)

The fundamental approach in DCGANs is to add upsampling convolutional layers between the input vector z and the output image in the generator. In addition, the discriminator employs convolutional layers similar to a conventional CNN or convolutional neural network. The architecture is presented in Figure 6. There are some notable changes in the architecture of DCGAN compared to the original FCGAN, which benefits high-resolution modeling and more regular training:

1) DCGAN replaces all pooling layers with strided convolutions in the discriminator and fractional-strided convolutions in the generator.
2) Batch normalization is used for both the discriminator and the generator, whereby similar statistics can be found for the artificial and authentic examples.
3) ReLU activation in the generator is used in all layers except the output using Tanh, while LeakyReLU activation in the discriminator is used for all layers.

Subsequently, the LeakyReLU activation prevents the network from getting stuck in a "dying state" situation (e.g., inputs less than 0 in ReLU) when the generator receives gradients from the discriminator.

DCGAN is trained on Large-scale Scene Understanding (LSUN) [35] and ImageNet datasets using 64×64-pixel images. This model was trained using stochastic gradient descent (SGD) with a mini-batch size of 128. Additionally, All weights were initialized from a Normal distribution with a mean of zero and a standard deviation of 0.02. Furthermore, the paper used hyperparameter tunning on the surface level parameters such as learning rate in Adam optimizer with an initial value of 0.0002 and a momentum term of 0.5. Finally, the slope of LeakyReLU was set to 0.2 for this model.

Moreover, DCGAN is a significant milestone in the GANs history where convolution becomes the prominent architecture used in the generator. However, since the model capacity and the optimization used in DCGAN are limited, it only thrives on low-resolution and less diverse images.

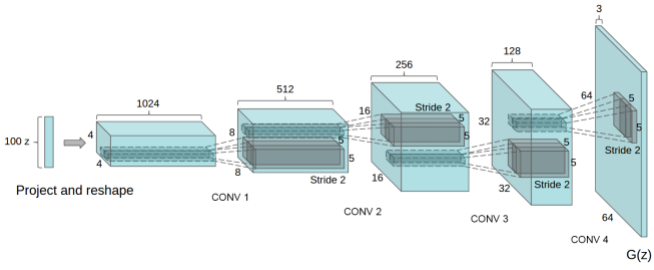

Fig. 6. DCGAN Architecture (from [34])

G. Self-attention GAN (SAGAN)

Conventional CNNs can only detect and learn local spatial features, and the target domain data may fail to cover adequate structure. This trait causes CNN-based GANs to struggle to learn multi-class image datasets such as ImageNet, and the critical components in generated images can be misinterpreted or shifted; for instance, the nose in a face-generated image may not appear in the correct position. Therefore, self-attention mechanisms have been proposed to guarantee a substantial receptive field without compromising computational efficiency for CNNs [32]. This mechanism in the GAN framework has been implemented for both the discriminator and generator architectures.[33] With this tool, SAGAN can learn global, far-reaching dependencies for image generation. As shown in Figure 7, the fundamental idea of this implementation is the attention map resulted from a softmax distribution, defining which features are more prosperous and better to succeed in more reliable performance of the final task, in this case, the image generation.

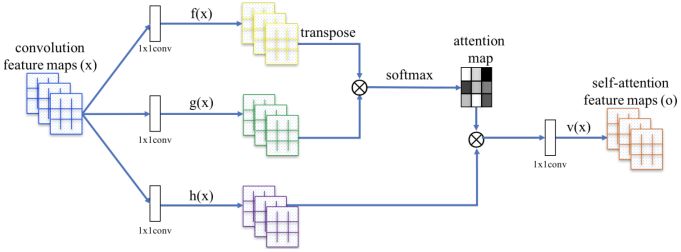

Fig. 7. SAGAN Architecture (from [33])

The chart from the original paper, represented in Table I, shows where these attention blocks (the mechanism) have been put and the results corresponding to the selection. It also points that SNGAN has outperformed the residual ResNet skipped connection block in this experiment in all options.

Model	no attention	SAGAN				Residual			
		$feat_8$	$feat_{16}$	$feat_{32}$	$feat_{64}$	$feat_8$	$feat_{16}$	$feat_{32}$	$feat_{64}$
FID	22.96	22.98	22.14	**18.28**	18.65	42.13	22.40	27.33	28.82
IS	42.87	43.15	45.94	51.43	**52.52**	23.17	44.49	38.50	38.96

TABLE I
DERIVED FROM [33]

SAGAN has delivered high performance on multi-class image generation by being trained on the ImageNet dataset with 128×128-pixel images. For implementation purposes. Spectral normalization was used in both D and G. Conditional batch normalization was used in the generator, while batch projection was used in the discriminator. In addition, In addition, Adam optimizer, and imbalanced learning rates was applied for D and G ($D_{LR} = 0.0004$ and $G_{LR} = 0.0001$).

Furthermore, as shown in Table I, the self-attention mechanism deployment for both the discriminator and the generator deployment self-attention mechanism at feature map with 32 × 32 sizes achieves the best FID score, and at feature map with 64 × 64 sizes achieves the best Inception score. This result indicates that the self-attention mechanism is complementary to convolution for large feature maps and improves the diversity for GANs.

Batch	Ch.	Param (M)	Shared	Skip-z	Ortho.	Itr $\times 10^3$	FID	IS
256	64	81.5	SA-GAN Baseline			1000	18.65	52.52
512	64	81.5	✗	✗	✗	1000	15.30	58.77(±1.18)
1024	64	81.5	✗	✗	✗	1000	14.88	63.03(±1.42)
2048	64	81.5	✗	✗	✗	732	12.39	76.85(±3.83)
2048	96	173.5	✗	✗	✗	295(±18)	9.54(±0.62)	92.98(±4.27)
2048	96	160.6	✓	✗	✗	185(±11)	9.18(±0.13)	94.94(±1.32)
2048	96	158.3	✓	✓	✗	152(±7)	8.73(±0.45)	98.76(±2.84)
2048	96	158.3	✓	✓	✓	165(±13)	8.51(±0.32)	99.31(±2.10)
2048	64	71.3	✓	✓	✓	371(±7)	10.48(±0.10)	86.90(±0.61)

Fig. 8. Results of BigGAN. Adopted from [36].

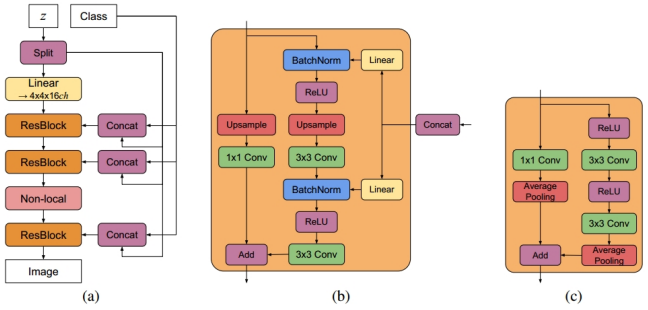

Fig. 9. (a) Architecture of generator in BigGAN. (b) A Residual Block in generator. (c) A Residual Block in discriminator. Adopted from [36].

H. BigGAN

BigGAN [36] is scaled-up model of SAGAN which uses some operations to enhance the capability of the model. The effect of these operations is shown in Fig. 8. In BigGAN, class labels are provided to generator using class-based normalized batches. Furthermore, moving average of generator's weights with a decay of 0.9999 of across training iterations increased stability of training process. Moreover, discriminator is updated twice in each iteration. Despite previous works which used Xavier initialization or N(0,0.02), Orthogonal Initialization is used to avoid exploding and vanishing gradients. As can been observed in Fig. 8, a remarkable impact is observed for increasing batch size so that enhancing the batch size from 256 to 2048 has enabled the model to improve its IS by 46%. Another performed scale-up is enhancing the number of channels in each layer by 50% which lead to increase IS by 21%. In addition, using shared embeddings, and skip-z connections had a significant impact on the performance. To control trade-off between image fidelity and variety, truncated Gaussian is utilized during applying the model. BigGAN is trained on ImageNet dataset with three type of resolutions include of 128 × 128, 256 × 256, and 512 × 512 and the operations improved the performance and BigGAN has reached state-of-the-art performance.

I. Your Local GAN (YLG)

Your Local GAN [37] retains locality and geometry using a new presented local sparse attention layer which supports information through attention steps. The authors replaced dense attention layer of SAGAN with new construction and reached highly remarkable FID, Inception Score, and pure visual enhancements so that FID score was increased from 18.65 to 15.94.In YLGAN, the main used idea is an approach which is called ESA (Enumerate, Shift, Apply). Indeed, All of the pixels are enumerated according to Manhattan distances from the pixel at position (0,0). Then, indices on one-dimensional sparsity are shifted to match the Manhattan distance enumeration, and finally, this new one-dimensional sparsity is applied. Furthermore, an innovative loss function is made by utilizing the discriminator's attention layer. This idea helped authors to show the ability of their model in capturing essential features of authentic images. Another proposed technique solves inversion problem of GAN by utilizing discriminator's attention to compute importance of pixels and also as a representation inversion loss. Nevertheless, despite the stated advantages of this GAN model, there exist some conflicts. Their approach makes the networks sparse while still full information flow is needed.

IV. Summary

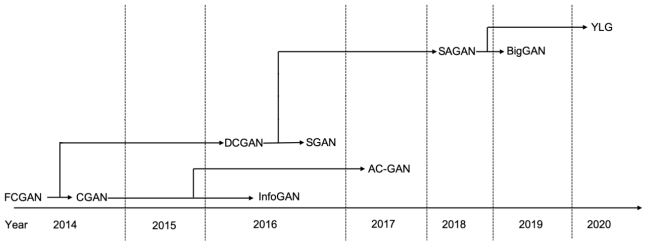

Fig. 10. GAN Architectures from the year 2014 to 2020

Figure 10 displays some GAN architectures from 2014 to 2020 derived or inspired from the other GAN variants. We have presented an overview of architecture-variant GANs, which strive to improve performance based on the two chief challenges: Image quality and Model diversity. Therefore in this section, we will recap each one of these objectives based on the discussed GANs. In addition, we will discuss some challenges that GAN architectures have faced in other domains.

In terms of image quality, one of the primary purposes of GANs is to produce more realistic images, which requires training on high-quality image datasets. The original GAN (FCGAN) is only applied to MNIST, Toronto face dataset, and CIFAR-10 due to its limited capacity of architecture. DCGAN introduced convolutional and up-sampling layers in the architecture and some "deep learning-based" modifications such as hyperparameter tuning. These implementations enable the model to have a more prominent capacity to generate higher resolution images. Other architecture variants such as SAGAN and BigGAN all have some adjustments on their loss functions, enhancing the image quality of the produced samples.

Furthermore, model diversity is the biggest problem for GANs. It is not easy for GANs to create realistic diverse images like natural images. With architectural-variant GANs, only SAGAN and BigGAN address such conflicts. The self-attention mechanism supports CNNs in SAGAN and BigGAN

to process large receptive fields, thereby overcoming the problems of shifting the components in the generated images. This process enables such types of GANs to produce more diverse data.

GANs have initially been proposed to produce believable synthetic images and have achieved impressive performance in the computer vision area. GANs have been applied to other fields, such as time series generation and natural language processing, but resulted in relatively fewer achievements. Compared to computer vision, Gans's research in other areas is still somewhat limited. These limitations are caused by the differences between the image and non-image data and the properties that make the available architectures more of a single-purpose model for solely image applications. For instance, GANs produce continuous value data, but natural language is based on discrete values like words, characters, and bytes, so it is laborious to apply GANs for natural language applications. As this field is becoming more thriving nowadays, more GAN architectures are trying to cover various types of data to be generated adequately and sufficiently.

V. Conclusion

Generative adversarial learning is one of hottest topics in unsupervised learning. A wide range of researches are performed to improve and employ GAN models in lots of domains. In this paper, a comprehensive review of various concepts of GAN is presented. Moreover, the most useful architectures and their applications are presented which provide a comprehensive sight about utilizing GAN in applied problems. We hope that this study help researchers to gain a thorough understanding of GAN and open research areas.

References

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

[2] J. Kossaifi, L. Tran, Y. Panagakis, and M. Pantic, "Gagan: Geometry-aware generative adversarial networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[3] Q. Dai, Q. Li, J. Tang, and D. Wang, "Adversarial network embedding," *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[4] N. Kodali, J. D. Abernethy, J. Hays, and Z. Kira, "How to train your DRAGAN," *CoRR*, vol. abs/1705.07215, 2017.

[5] A. Borji, "Pros and cons of GAN evaluation measures," *CoRR*, vol. abs/1802.03446, 2018.

[6] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Q. Weinberger, "An empirical study on evaluation metrics of generative adversarial networks," *CoRR*, vol. abs/1806.07755, 2018.

[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *CoRR*, vol. abs/1704.00028, 2017.

[8] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a nash equilibrium," *CoRR*, vol. abs/1706.08500, 2017.

[9] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, 2012.

[10] Z. Wang, G. Healy, A. F. Smeaton, and T. E. Ward, "Use of neural signals to evaluate the quality of generative adversarial network performance in facial image generation," *CoRR*, vol. abs/1811.04172, 2018.

[11] Z. Wang, Q. She, A. F. Smeaton, T. E. Ward, and G. Healy, "Synthetic-neuroscore: Using a neuro-ai interface for evaluating generative adversarial networks," *Neurocomputing*, vol. 405, 2020.

[12] S. Barratt and R. Sharma, "A note on the inception score," 2018.

[13] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," 2016.

[14] K. Kurach, M. Lucic, X. Zhai, M. Michalski, and S. Gelly, "The gan landscape: Losses, architectures, regularization, and normalization," 2018.

[15] S. Hitawala, "Comparative study on generative adversarial networks," 2018.

[16] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, 2017.

[17] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, 2018.

[18] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial networks and their variants work: An overview," *ACM Comput. Surv.*, vol. 52, 2019.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[21] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudík, eds.), vol. 15 of *Proceedings of Machine Learning Research*, pp. 315–323, 2011.

[22] A. Odena, "Semi-supervised learning with generative adversarial networks," 2016.

[23] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014.

[24] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," 2016.

[25] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3d face model for pose and illumination invariant face recognition," 09 2009.

[26] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic, "Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models," in *CVPR*, 2014.

[27] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[28] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," 2017.

[29] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, 05 2012.

[30] C. Rabinovitz, N. Grupen, and A. Tamar, "Unsupervised feature learning for manipulation with contrastive domain randomization," 2021.

[31] A. Spurr, E. Aksan, and O. Hilliges, "Guiding infogan with semi-supervision," 2017.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[33] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2019.

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

[35] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," 2016.

[36] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," *CoRR*, vol. abs/1809.11096, 2018.

[37] G. Daras, A. Odena, H. Zhang, and A. G. Dimakis, "Your local gan: Designing two dimensional local attention mechanisms for generative models,"