SURVEY: GANS FOR DATASET GENERATION

[1] Supreetha M, [2] Dr. Anala M R
[1]Student, [2]Associate Professor
[1]Computer Science and Engineering
[1]R V College of Engineering, Bangalore, India

Abstract : - Large amount of data is essential for any machine learning tasks for accurate and efficient results. But data may not always be available in huge numbers according to the requirement like medical data, aerospace data, and forensic data and so on. For this purpose, Generative Adversarial Networks are used which are a type of deep neural networks used for generating realistic datasets. It consists of two networks one the Generator and other the Discriminator whose task is to generate data samples and to differentiate between true and false sample respectively. This paper gives an overview of variants of GANs. Each type of GAN is trained differently according to the application of use, from which data can be generated and can be utilized for different purposes.

***IndexTerms* - GANs, CGANs, BCGANs, MedGANs, Wasserstein GANs, Water GANs**

I. INTRODUCTION

Artificial Neural Networks are biologically inspired paradigm. The general idea of neural networks is to behave in the same way as that of the human brain. This seems to be new concept but has been existed before the invention of computers. They are different in their structure and learning process. The main structure of neural networks consists of a set of large number of connected units called the artificial neurons which works together as one element with the aim to solve a problem. Figure 1 shows the architecture of neural networks where each circular node represents a single artificial neuron and each arrow in the figure is the connection between the two ends of the neuron that is from output end to the input end. Signal can be transmitted from and to each of the connected neurons. Each neuron has many number of inputs and only one output. It mainly has two modes of operation. The first mode of operation is the training mode and the second one is the using mode. In training mode, the neuron can be trained as to when to fire and when not to fire based on the patterns of input. In using mode, if the input is a trained one to fire then its associated learned output will be actual output. If it is not a learned input then a firing rule is used to decide if it should be fired or not to be fired. Firing rules are a very important in neural networks as it is accountable for its flexibility. It is the one which determines if a neuron is to be fired for an input. This enables the neurons to respond similarly and sensibly to same kind of inputs during training phase. Each neuron that will receive the signal will process the signal and signal it to another neuron that is connected to it. This signal is generally calculated by the sum of all the inputs using a function which is non-linear in nature. Each connection have a weight that is assigned to it and as the learning process goes on it adjusts the weight according to the strength of the signal.

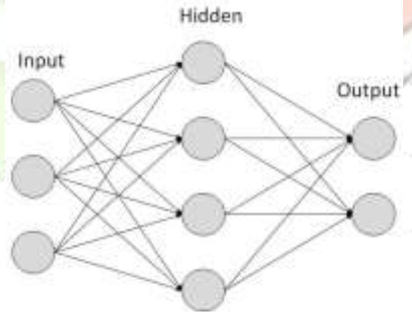

Figure 1 Architecture of Neural networks

Bose in 1996 classified neural networks into recurrent and non-recurrent neural networks. Recurrent networks are the networks which involves feedback and non-recurrent networks are the one which do not have the feedback mechanism. The main and most important applications of neural networks is in the field of recognition of patterns and speech, machine translation, in medicine and so on.

II. RELATED WORK

Generation of images in the recent has been a very important problem. Images mainly consists of many number of objects which forms a structure that are hierarchical in nature which has different shapes and different styles. Deep learning has been widely used in generating images. As large number of dataset are not available in all the fields and availability of large data is very essential for many classification problems which have mainly focused on the finite number of labels which needs concrete information. This is not available in the situation of discriminative models. This triggers the challenges that will occur in the task of generation which is the main problem of artificial intelligence.

In [1] the author has explained about a type of Generative Adversarial Networks that has the capability of avoiding the over fitting of data. This over fitting generally occurred due to the complexity of exponential of images. These images may have many number of objects in a single image which has noisy features like the size, the color and the shape. Composite Generative Adversarial Network (CGAN) had the ability to generate all the images individually part by part instead of generating the entire image in a single shot. They are different from that of the recurrent neural networks where it blindly adds the images in a sequence by overlapping them thereby blurring the images. This has been overcome in CGAN which makes use of process called alpha blending. It uses an alpha channel in conjunction with the RGB channels to iteratively stack the images with the help of this blending process. In process of alpha blending the previously formed image will be preserved in certain areas and then it overlaps the newly formed image perfectly in the other areas. In the first stage, the Recurrent Neural Networks will accept the noise vector z as the input for the process. The value will be again fed to the next loop at each t time step. Recursively the inputs are passed through each generator by which it creates images which are having channels of RGB then sequentially images are combined by the alpha blending process to form the final image and each generator used will be different.

The existing methods for extraction of road images uses binary classification because of the eminent performance of the classification methods. But in [2] an end to end GAN was proposed. In general, high precision images are required for road detection which are obtained from remote sensing and it is of use in variety of applications. Many approaches do not have the capability of extracting the features of the road with high accuracy. A convolution neural network is implemented where it is given training in an adversarial manner. This will be different from that of segmentation maps that is obtained from the ground truth. The images that are generated from the segmentation model and convolution neural network was able to improve the image by correcting the difference between the both the images. This model of GAN can outperform other existing methods based on the performance.

In [3] a MalGAN was designed which had the capacity of generating examples for attacking on algorithms that were designed for detection of blackbox malware. The architecture consists of substitute detector which is used to fit this detection system. A generator neural network is trained to minimize the malicious probabilities that are predicted by the substitute detector for the adversarial samples that are generated. This was used to generate malware samples and were able to minimize the rate of detection close to zero and was able to make a defensive model.

In [4] a Deep Recurrent Attentive Writer specifically called the DRAW neural network which is used for image generation is discussed. This architecture combines the attention mechanism and the variational auto encoding framework. Attention mechanism tries to mimic the way how the human eye would do and the variational auto encoding is used for the repetitious construction of the generated images. It consists of an encoder that has the capability of capturing input data's silent information and a decoder will receive the data and will condition them over its own distribution of images. Both the networks that is the encoder and decoder used here are recurrent networks. Hence a sequence of samples can be exchanged and the output of the decoder will be consecutively added. The added samples to the distribution will be then used to generate the data. At each time step the network makes a decision as to from where the data has to be read and from where and what data should be written. This mechanism has the capability of generating images very similar to the original ones which will be indistinguishable by the human eye.

III. GENERATIVE ADVERSARIAL NETWORKS

Variational Auto Encoders (VAEs) and GANs are best known for their property of generating samples. VAE mainly focusses on the likelihood and have a limitation that they needed to fiddle with the noise terms that were added additionally. In respect to this, GANs have much more flexibility in the objective function like Jensen-Shannon, f-divergences and exotic combinations. Generative Adversarial Networks are a new concept of neural networks introduced by Ian Goodfellow [5] and his co-workers in 2014. To understand GANs, two concepts need to be known. One is the types of machine learning algorithms and other types of models. First the two types of machine learning algorithms are supervised and unsupervised learning. In supervised learning the machine will be trained with set of labeled data and it tries to learn from that set of sample data and for the rest of the data the machine will be automatically trained based on the trained data sample. It is mainly used for large quantities of labelled data. In case of unsupervised learning where the machine will not be trained but they have to learn on their own that is they learn from their mistakes and makes sure that they don't repeat that mistake again. One of the biggest disadvantage of the supervised learning is that they require large amount of labeled data but this is not available in the real scenario and it may be expensive and time consuming as well. Whereas this disadvantage has been overcome in the case of unsupervised learning but is less accurate. Generative Models is the one that have the capability of generating samples for given sample of inputs. By having these two concepts GANS could be easily visualized.

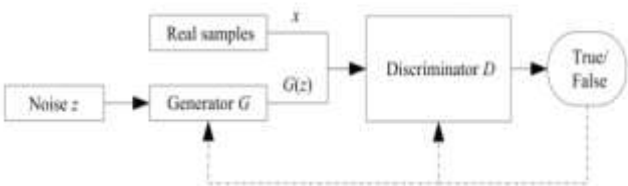

Figure 2 Architecture of GAN

As the name suggests that the samples are generated in adversarial manner by using the concepts of neural networks. It is the architecture based on unsupervised learning. It has a better performance when compared to the traditional neural networks. GANs have two nets that have the capability of working independently and are the adversaries in the architecture. The Figure 2 represents the architecture of GAN. First network is the Discriminator (D). D is the net that has to undergo vigorous training phase so that it can understand about the samples that will be fed to it and to make correct prediction. Figure 3 shows the training stages of GAN. The second net is the Generator (G) that will generate fake samples that is similar to that of the real one. The idea of GANs is like two player game where one player is G and the other player is D. To win this game each of these networks will try their best to optimize in order to upgrade their generation and discrimination talent and to find out the Nash equilibrium that exists between both the networks. G and D makes usage of a differentiable function. The input to the generator are the original sample(x) and the random variable (z). Both of them are fed to the generator, the task of generator is to generate a fake sample from the inputs provided with the aim of fooling the discriminator. Thus the output image obtained from Generator is G(z). Now D will be fed with this sample (G(z)) and with the original list of samples. The task of discriminator is to identify the true and the fake sample from the set of samples provided to it. If D correctly identifies the sample as original or fake then G tries to improve itself to produce better samples else it fails to identify. Then it is task of D to avoid such kind of errors further. This is given by 0s and 1s. 1 for true D(x) and 0 for false D(G(z)). Reward of D is the correct number of predictions and for G it is the number of errors that is made by D. This mechanism continues until equilibrium is achieved. That is at this point D can correctly identify the sample. The optimization of adversarial function gradually improves the performance of D and G. During the training phase, the parameters of D are updated 'k' times and then the generator is updated once. The optimization of D is given by G and loss function is defined in equation 1.

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

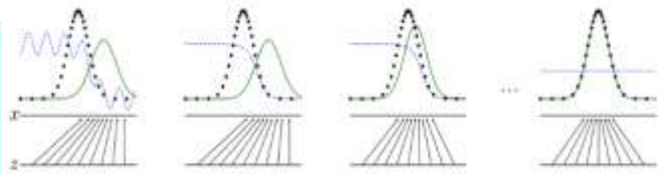

Figure 3 Training stages of GAN.

Dark dotted line represent the true data. Solid line represents the generated data. Light dotted line represents the discriminator loss. In equation 1, the function V(D,G) entropy is the first term in which the data is taken from the read distribution that is $p_{data}(x)$. The aim is to minimize the value of G and maximize the value of D in the loss function. Log(D(x)) represents the discriminators capability to identify the data as being real and 1-log(D(G(x))) is the discriminators capability to classify the samples generated by the generator to be fake.

IV. TECHNICAL RELEVANCE

The proposal of GAN by Ian Goodfellow led to the development of various variants on GAN. The data may not be always available. In some cases the data may be available but may be very scarce which may be insufficient for processing the data for analysis. Hence there is a need for huge amount of data. GAN are a very good source for generating data. From a small set of images it can generate many number of images

4.1 Conditional Generative Adversarial Networks

GANs concepts can be extended by adding a condition that is an external information on both the discriminator and the generator called the Conditional Generative Adversarial Networks(CGAN). This condition can be a vector y that is a class label or any data based on which the image will be conditioned and the new image will be generated. This y will be fed as an additional input. Now the generator will have three inputs one the sample data (x), noise (z) and the condition (y). z and y will be combined by some joint hidden representation. Flexibility will be ensured by the adversarial training framework. The architecture of CGAN is shown in Figure 4. This model of GAN enables to work in different modes by giving them various contextual informational for each data that is generated. The actual variation of CGAN loss with that of the original GAN loss lies in the conditional vector that is fed to both the generator and the discriminator. The equation 2 gives the objective function of CGANs where G and D are conditioned over a variable y. [6]

$$\min_G \max_D V(D,G) = E_{z\sim p_{data(x)}}[logD(x|y)] + D(G(z|y)))] \quad (2)$$

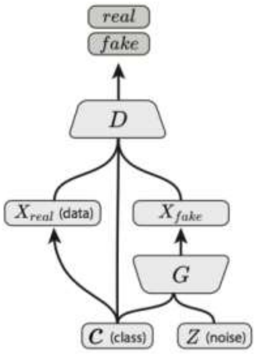

Figure 4 Conditional GAN

In the generator part, the noise and the conditional data are combined to create a dense code of the output image. The G that is used here will be a deconvolutional neural network where the input is run over filters upsamples the samples rather than contracting them into a higher dimensional space representation. Only one deconvolution is used here and thus helps in preventing the over fitting the data that is to be trained. The discriminator behaves as a convolution neural network in which each layer has a maxout activation and the final output will be a dense code. An important decision needs to be taken as to where to insert y. The information that is being used has to take the decision about the data generation or to make a boundary that helps for the decision purpose by the discriminator. This will be either added at the start of the feedforward process that is before deconvolution and other at the end of the feedforward process after the convolution. This dense code and the conditional data are again combined together to yield the prediction. The prediction is generally done by using the binary classifiers that either 1 or 0.

4.2 Wasserstein GANs

Wasserstein GANs[7] is a type of GAN that can minimize and efficiently approximate the Earth Mover (EM) distance. EM distance is the distance between the two probability distributions. It is also called the Wasserstein Distance. It informally can be explained as moving the large amount of dirt that exists in a probability distribution at the lowest cost such that it can resemble the other distributions. The EM distance is a differentiable and continuous which means that training the critic can be done until optimality. Because of these properties the discriminator can be strongly trained before the generator is being updated which enabled it to receive better and improvised gradients from the discriminator to train itself. GANs make use of this distance as the loss function. Another advantage of WGANs is that the problems in training the GANs are cured. They need not maintain a balance while training the generator and the discriminator and as well as need not require a proper design of the architecture. The dropping phenomenon in the mode is also dramatically reduced. Thus optimality is achieved. By plotting the learned values is useful while debugging and for hyperparameter searches and also they correlate well with the quality of the observed samples. WGAN loss function well correlates with the quality of the samples, performs well and produces good images. Even if the batch norm present in generator is removed it produces good samples but others models fail when this is done. They avoid mode collapses when compared to the standard GANs and uses a feedforward network rather than traditional convolution network and thus the number of parameters will remain the same thus removing the inductive bias which was present in the convolution networks.

4.3 Bidirectional Conditional GAN

Another type of GAN called the Bidirectional GAN formally called the BiGAN[8] also called the Adversarially Learned Inference, ALI model which is framework that has recently developed. In this model the encoder is made to learn the inverse mapping from the noise data to the sample of data distribution for latent distribution but not reverse mapping that is from data to the noise. This mapping is very essential as it gives an information rich and compact representation of the data that can be used as inputs for the classification in an effective way. The encoder is trained along with the discriminator and the generator. By combining BiGANs and the Conditional GANs (cGANs) in which it extends the conditional GANs in a bidirectional setting called the Bidirectional Conditional GAN (BCGAN) [9]. The encoder is trained along with the CGAN and the inverse mappings for both extrinsic and intrinsic factors are learned for the samples of data. Thus BCGANs can inherit the mode coverage, regularization and robustness against mode collapse that occurred in BiGANs. If BCGANs are trained in the way as the BiGANs they do not give good results. BCGANs are trained by which encoder can do the inverse mapping of extrinsic attributes and such that generator can use them while generating samples.

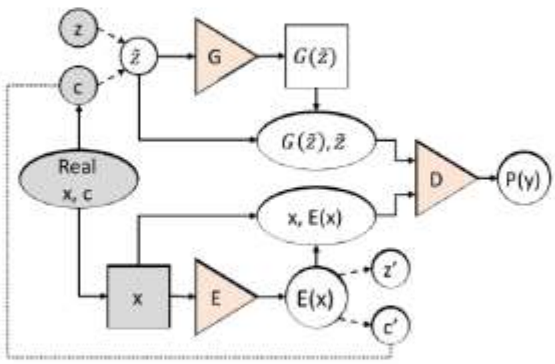

Figure 5 Bidirectional Conditional Generative Adversarial Network

The min max objective for BCGANs is represented in equation 3.

$$\min_{G,E}\max_{D} V(D,G,E) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(E(x),x)] + \gamma\, \mathbb{E}_{(x,c)\sim p_{data}(x,c)}[\mathrm{EFL}(c, E_c(x))] + \mathbb{E}_{\hat{z}\sim p_{\hat{z}}(\hat{z})}[\log(1 - D(\hat{z}, G(\hat{z})))] \tag{3}$$

As depicted in figure 5[9], the inverse mappings are done from z to x and also to c unlike which the generator should also learn so that it could also fool D. From the latent distribution, the intrinsic factors can be randomly sampled but for extrinsic factors it needs a specialized and specific high level of information which was the difficulty in BiGANs and was overcome by adding extrinsic factor loss shortly called EFL, an explicit mechanism to encode the extrinsic factors. In this each c associated with x can be accessed and can be used for learning the inverse mapping from data to the conditional vector. Only adding this doesnot produce best results as there may be backpropogation gradients to EFL. In order to avoid the difficulty in modeling the distribution, EFL is multiplied by importance weight ɣ which is called the importance weight (EFLW).

4.4 MedGAN

Data is an important part while doing any analysis and predictions but access to data in some fields is very critical because of security reasons, ethical and legal matters. Anonymization can be done so that the personal details can be deidentified[10]. But anonymized data will not perform well. So MedGAN is a Medical GAN used for generating images related to medical and biomedical field. Images were generated based on images obtained from Electronic Health Record, EHR. Figure 6 represents the MedGAN architecture. In the figure 6[10], x is the sample from EHR, z is the noise randomly specified to the generator G. The generator is a feed forward network which has short connections. It uses an encoder and a decoder which learned the samples x. The autoencoder are given training so that the samples can be represented in a lower dimensional space and then back to original dimensional space. The decoder will be reused again after G inorder to learn the features to reconstruct the discrete output. Now discriminator D will try to identify the real sample and the output obtained from Dec(G(z)).[11]

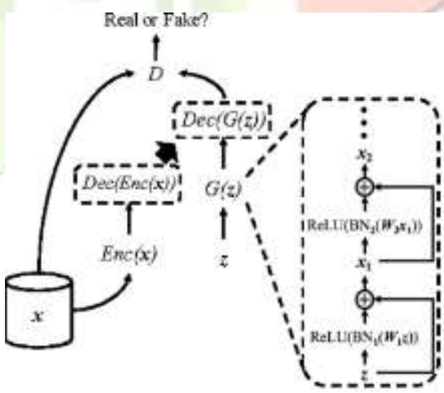

Figure 6 MedGAN Architecture

4.5 WaterGAN

WaterGAN is a type of GAN used for realistic generation of under water images. The cameras and other vehicles that operate remotely can be used capture images of high resolution. Underwater images formed will be subjected to complex process of light propagation. The images that are obtained will be different from the original one as they would have undergone through various phenomenons of light like scattering, absorption which results in attenuation of light based on different wavelengths. Restoring of these kind of under water images become difficult by physical process so using WaterGAN[12] large set of images can be generated and the produced image will be given as input to the 2 stage network mainly used for the color correction of the images.

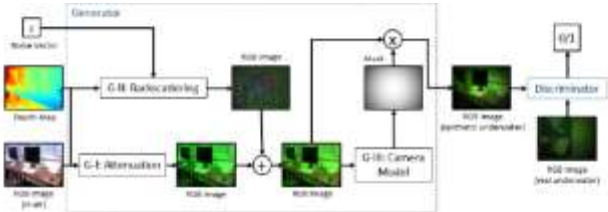

Figure 7 WaterGAN

The generator has 3 stages as shown in figure 7[12]. They are attenuation, backscattering and camera model. The main aim of having three stages in WaterGAN is to align the generated images with the RGB-D input so that at each stage it doesnot alter the structure rather focuses on the color and intensity. Network uses the depth information as it necessary for the discriminator which has no knowledge about the depth. The input and output images are in the same resolution. To maintain this aspect ratio, scattering image and vignetting mask are upsampled with the help of bicubic interpolation before they are applied to the sample images.

V. APPLICATIONS OF GAN

GANs are widely used in image and in vision computing. SRGAN is a type of GAN presented mainly for the super resolution of images. This GANs try to enhance the resolution of images by getting the textual details[13]. BEGAN can be used to generate face samples of high quality with varied expressions, skin color, poses, genders, hair texture and so on. SimGAN [14] are a type of GANs that have the capability to learn from unsupervised and simulated images for reducing the gap between real and synthetic image data. GANs are also used for image to image translations in which it does a mapping between input and output images. In absence of example pairs, CycleGANs [15] are used to translate a source image to a target image. They can be used for style transfer, attribute transfer, photo enhancement and object transfiguration. GANs can also be used for capturing the dialogue and generating its corresponding text. SeqGAN [16] use reinforcement learning in order to generate language and speech, music and poems. STGANs [17] Style Transfer GANs are used for transferring styles. It is a way of learning chess in a different way based on the style of specific individual. Generic GANs are also used for image inpainting where some portion of the image may not be available and be filled with the help of GANs. GANs can also generate videos, prediction of next sequence of frame, image to text translation. GANs may also be used for inverse reinforcement learning and imitation learning like for imitating the human drivers to achieve the goals of self driving cars.

VI. CONCLUSION

Generative Adversarial Networks are one of the successful technique in unsupervised learning that has the capability to implement the generative tasks. GANs have gained popularity in generating realistic data from the data samples of high dimensionality of certain distribution and use the data for prediction and analysis especially in case where the size of data is very small. This is the only model that can perform in supervised, unsupervised and even semi-supervised learning. There is a lot of research going on to improve the training methods, increasing their stability and so on.

REFERENCES

[1] Hanock Kwak, Byo5ung-Tak Zhang, "Generating Images Part by Part with Composite Generative Adversarial Networks", arXiv:1607.05387v2 [cs.AI], 2016

[2] Shi Qian, Liu Xiaoping, Li Xia, "Road Detection from Remote Sensing Images by Generative Adversarial Networks", IEEE, Issue: 99, 2017

[3] Weiwei Hu, Ying Tan, "Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN", arXiv: 1702.05983v1 [cs.LG], 2017

[4] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, Daan Wierstra, "DRAW: A Recurrent Neural Network For Image Generation", arXiv: 1502.04623v2 [cs.CV], 2015

[5] Kunfeng Wang, Member, Chao Gou, Yanjie Duan, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang, "Generative Adversarial Networks: Introduction and Outlook", IEEE/CAA Journal Of Automatica Sinica, Vol. 4, No. 4 pp. 588 - 598, 2017

[6] Mehdi Mirza, Simon Osindero, "Conditional Generative Adversarial Nets", arXiv:1411.1784v1 [cs.LG], 2014

[7] Martin Arjovsky, Soumith Chintala, and Leon Bottou, "Wasserstein GAN". arXiv:1701.07875v3 [stat.ML], 2017

[8] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, "Adversarially Learned Inference" arXiv:1606.00704v3 [stat.ML], 2017.

[9] Ayush Jaiswal, Wael AbdAlmageed, Yue Wu, Premkumar Natarajan, "Bidirectional Conditional Generative Adversarial Networks", arXiv:1711.07461v1 [cs.LG], 2017

[10] Sharmilan S, Hapugahage Thilak Chaminda,"Generate bioinformatics data using Generative Adversarial Network: A Review ", 2nd International Conference on Information Technology Research, 2017

[11] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, Jimeng Sun, "Generating Multi-label Discrete Patient Records using Generative Adversarial Networks", arXiv:1703.06490v3 [cs.LG], 2018

[12] Jie Li, Katherine A. Skinner, Ryan M. Eustice, and Matthew Johnson-Roberson, "WaterGAN: Unsupervised Generative Network to Enable Real-time Color Correction of Monocular Underwater Images", IEEE Robotics and Automation Letters, Volume:3, Issue:1, pp. 387 - 394, Jan. 2018

[13] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. H. Wang, and W. Z. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," arXiv: 1609.04802, 2017.

[14] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: boundary equilibrium generative adversarial networks," arXiv: 1703.10717, 2017.

[15] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," arXiv: 1703. 10593, 2017.

[16] L. T. Yu, W. N. Zhang, J. Wang, and Y. Yu, "SeqGAN: sequence generative adversarial nets with policy gradient," arXiv: 1609.05473, 2016.

[17] M. Chidambaram and Y. J. Qi, "Style transfer generative adversarial networks: Learning to play chess differently," arXiv:1702.06762,2017