Audio Enhancement and Synthesis using Generative Adversarial Networks: A Survey

Norberto Torres-Reyes
Department of Electrical and Computer Engineering
University of Nevada, Las Vegas

Shahram Latifi
Department of Electrical and Computer Engineering
University of Nevada, Las Vegas

ABSTRACT
Generative adversarial networks (GAN) have become prominent in the field of machine learning. Their premise is based on a minimax game in which a generator and discriminator "compete" against each other until an optimal point is reached. The goal of the generator is to produce synthetic samples that match that of real data. The discriminator tries to classify the real data as real and the generated data as not real. Together, the generator improves to the point where the fake data and real data are identical to the discriminator. GAN has been successfully applied in the image processing field over a large range of GAN variant architectures. Although not as prominent, the audio enhancement and synthesis field has also benefitted from GAN in a variety of different forms. In this survey paper, different techniques involving GAN will be explored relative to speech synthesis, speech enhancement, music generation, and general audio synthesis. Strengths and weaknesses of GAN will be looked at including variants created to combat those weaknesses. Also, a few similar machine learning architectures will be explored that may help achieve promising results.

General Terms
Generative Adversarial Networks, Survey, Audio Synthesis, Audio Enhancement.

Keywords
Audio, synthesis, generative adversarial networks, survey, enhancement.

1. INTRODUCTION
Generative adversarial networks (GAN) are a recent introduction to supervised and unsupervised machine learning. In 2014, Ian Goodfellow, et al introduced the concept of pairing a generator network along with a discriminative network in a minimax game [1]. The two networks compete until the optimal solution is met. The paper uses a detective and a forger as an example. As the detective gets better at recognizing fakes, the forger gets better at making them. The optimal solution is met when the detective cannot tell the difference between the real works of art and the fakes. Prior to GAN, success in deep learning has come from discriminative models and less so from generative models [1]. The original study uses a multi-layer perceptron for both the generator and the discriminator and is trained with back propagation. Random noise is fed into the generator which maps the noise to a generated data sample. The discriminator outputs a probability that the data it receives is from the real data distribution and not the generated data distribution.

Audio synthesis has many practical applications in all different types of industries [2]. Many practical applications include speech enhancement, sound effect generation, and music generation[2][3]. A method that has been used to synthesize audio is statistical parametric speech generation. Pitfalls of this method are that they may seek to reduce the mean square error between the real and synthesized speech. Nevertheless, a reduction in error doesn't necessarily mean a more realistic output to listeners [3]. Recurrent Neural Networks (RNN) have been used in speech enhancement and music generation using recursive operations which make it specifically well suited for learning sequences [4]. Due to the recursive nature though, they may not be as quick as with GAN which can be used in parallel with the raw audio [3]. Variational Autoencoders (VAE) have also been prominent and successful in the audio field. With VAE, the idea is to have the latent variables follow a Gaussian distribution, which the decoder will use to learn a mapping between the distribution and the examples [4].

This paper provides an overview of several studies related to audio synthesis and enhancement using generative adversarial network. First, it is necessary to explain what GAN is and what improvements have been made to the algorithm. Next is an overview of several GAN based architectures that have been successfully applied to the audio field. Some non-GAN alternatives are also mentioned that may be beneficial in future research.

2. PRELIMINARIES

2.1 What is GAN
A more formal definition of GAN can now be described following the original paper [1]. For simplicity, both the generator and the discriminator are said to be a multilayer perceptron. Parameters for the generator and discriminator are $\boldsymbol{\theta_g}$ and $\boldsymbol{\theta_d}$ respectively. The real data is $\mathbf{P_x}$**~X** and the input noise variables are $\mathbf{P_z}$**~Z**. The generator, **G**, maps **Z** to **X** to obtain generated data $\mathbf{P_g}$ and the discriminator, **D**, maps **X to [0,1]**. The goal is to maximize the discriminators probability of correctly labeling whether the data it receives came from **X** or not ($\mathbf{P_g}$ instead). At the same time, it is desirable for the generator to create samples that are good enough to be classified as real, thus fooling the discriminator. Both the generated data distribution $\mathbf{P_g}$ and the real data distribution $\mathbf{P_{data}}$ are used as the training set. [1] Shows that the global optimum of the algorithm is obtained when $\mathbf{P_g = P_{data}}$, which means that the value of the discriminator becomes **D(x) =** ½. At that point, the discriminator can no longer distinguish between the real data and the generated data. The minimax objective function can be seen below

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{data}(x)}[LogD(x)] + \mathbb{E}_{z\sim P_z(x)}[Log(1-D(G(z)))$$

The algorithm alternates between gradient ascent on the discriminator and gradient descent on the generator. The corresponding gradients of the discriminator and generator respectively are shown below:

$$\nabla\theta_d \frac{1}{m}\sum_{i=1}^{m}\left[\log\left(D(x^i)\right)+\log\left(1-\ D\left(G(z^i)\right)\right)\right]$$

$$\nabla\theta_g \frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G(z^i)\right)\right)$$

The most common issues seen with the GAN algorithm are a) vanishing gradients and b) mode collapse [1]. If the real data and the generated data have little overlap, the Jenson-Shannon (JS) divergence can saturate to a constant and cause vanishing gradients when using the gradient descent method of training [5]. Mode collapse is when the generator becomes concentrated near or at a single point and provides no meaningful result. This can occur because there is no explicit way for the generator to be forced to be diverse. Both these issues have been tackled and several approaches have been created to combat them.

2.2 Wasserstein GAN

Although GAN can perform well by itself, there have been numerous improvements and modifications[6][7]. One prominent modification that has been done is called WGAN (Wasserstein GAN) [5]. WGAN attempts to solve the issue with vanishing gradient. It does this by using the Wasserstein Distance instead of the JS divergence to calculate the difference between two probability distributions. This distance is also known as the Earth Mover's (EM) distance. The Wasserstein metric provides a smooth measure as opposed to the JS divergence and therefore is more stable when used with gradient descent [5]. The Wasserstein distance can be written as:

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

where $\mathbf{P_r}$ **and** $\mathbf{P_g}$ are the real data and generator distributions respectively. The EM distance can be transformed to become a GAN loss function where the discriminator instead assists in computing the Wasserstein distance. First, the above equation is transformed to instead compute the maximum value instead of the minimum value.

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

The function, **f** , is said to be K-Lipschitz continuous. To ensure that this continuity is maintained, weight clipping is used to maintain the weights within a small window. In [5], they use a window of -0.01 and 0.01. The WGAN loss function becomes:

$$L(p_r,p_g)=\max_{w\in W}\mathbb{E}_{x\sim\mathbb{P}_r}[f_w(x)]-\ \mathbb{E}_{z\sim p(z)}\left[f_w\left(g_\theta(z)\right)\right]$$

While WGAN is an improvement from GAN, there are still some issues to be addressed. Specifically, the weight clipping used to maintain K-Lipschitz continuity can result in slow convergence or even a failure to converge. [8] proposed that instead of a weight clipping, a gradient penalty can be used. The WGAN-GP then becomes:

$$L_{WGAN-GP}=\ \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})]-\ \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)]+\ \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}\left[\left(\left|\left|\nabla_{\hat{x}}D(\hat{x})\right|\right|_2-1\right)^2\right]$$

Where $\tilde{x}=G(z),\ z\sim p(z),\lambda$ is the penalty coefficient. Adding the gradient penalty to WGAN improves the overall performance and stability [1].

2.3 Deep Convolutional GAN

Another prominent improvement to the GAN architecture is the adaptation to deep convolutional networks, otherwise known as DCGAN (deep convolutional GAN) introduced by [9]. DCGAN aims to provide a stable GAN that can be used in unsupervised learning by applying a set of architectural constraints, and then using the trained GAN in later learning tasks. Main components of DCGAN include replacing pooling layers with strided convolutions in the discriminator and fractional-strided convolutions in the generator. Also, a batch norm is used in the both the discriminator and the generator. Rectified Linear Units (ReLUs) are used as an activation function in every layer of the generator except for the final output layer, which uses the tanh function. LeakyReLU is used in the discriminator for all layers.

3. PREVIOUS WORKS

The majority of GAN research has been done in the image processing field. The synthesis of photorealistic images has had a significant improvement since the implementation of GAN. Papers such as [10] have been able to reproduce high quality images using datasets such as the CelebA-HQ and the LSUN bedroom set. Other applications noted by [6][7] include image to image synthesis, super-resolution, classification and regression, and speech and language processing. Although the latter may not necessarily be audio synthesis, it does provide insight and techniques used to apply audio to GAN.

3.1 Speech Generation

After the success GAN has had with images, it is conceivable that speech and audio would be a natural extension [11][12] [13][3][14]. Speech generation and text-to-speech (TTS) systems sometime suffer from poor human perceptibility, or perceptual deficiency [3]. Other methods of applying GAN to speech and language processing have been used to generate sentences and poems or to generate text based on dialogue [7]. Statistical parametric speech synthesis (SPSS) has been successfully applied in this field but has had trouble when it comes perceptual quality. Paper [3] has attempted to address the issue by combining cGAN (conditional GAN) [15] and SPSS in a multi-task learning framework. This framework applies a mean squared error loss function along with the GAN loss function, as shown below.

$$Loss_{multi}=\arg\min_G\max_D\mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(D(x|y)\right)\right]+\ \mathbb{E}_{z\sim p(z)}[G(z|y)-X_{real}]^2+\ \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1-D(G(z|y)|y)\right)\right]$$

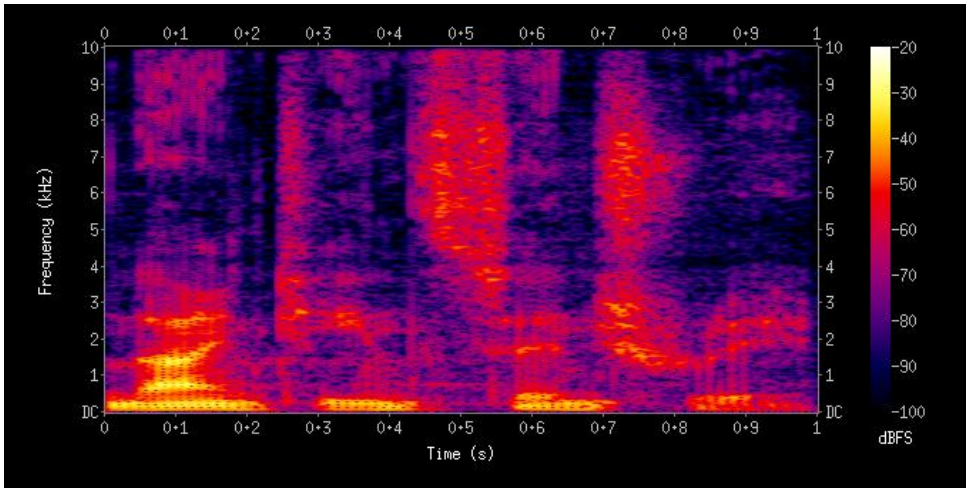

Figure 1: Spectrogram of spoken words. Courtesy of https://en.wikipedia.org/wiki/Spectrogram

Where X_{real} has probability distribution $p_{data}(x)$. The study tackles the issue of vanishing gradients by including the MSE loss function to improve stability. The results obtained by [3] showed that GAN can supplement SPSS by making up for the perceptual deficiency problem while only have a little increase in computational cost. Future works include optimizing the performance using a more stable form of GAN, such as WGAN [5]. Other recent studies [12] include improving speech in noisy environments using cGAN and spectrogram representations of audio. [11] focuses on the over-smoothing issue related to SPSS by creating a conditional GAN based post-filter to reconstruct a natural spectral structure in the synthesized speech. WGAN-GP was recently applied in [16] along with cGAN and WaveNet in a multi-speaker TTS synthesis system. The results showed that the WGAN-GP variant achieved the highest subjective evaluation score, as hypothesized by [3]. These several studies show that GAN can indeed improve synthesized speech quality and human perceptibility.

3.2 Speech Enhancement GAN

While many speech enhancement methods use spectrograms or SPSS methods, Speech Enhancement GAN (SEGAN) [17] operates on the waveform level. SEGAN can operate on raw audio and learn from different speaker and noise conditions. Speech enhancement in general takes in a noisy signal and enhances it. In this case, the generator performs the enhancement with the noisy inputs and a set of random variables and outputs the enhanced signal. The generator operation is fully convolutional, which is done to reduce training parameters and training time. The input signal is passed through several strided convolutions and parametric ReLUs. After decimation is completed, a thought vector **c** is concatenated with random variable **z**. Decoding of the signal follows a reversal of the encoding process. One key feature is the use of skip connections in which low level details of the signal pass straight through to the decoder [17]. The resulting loss function uses the L1 norm to help calculated the distance between the generated and actual distributions, resembling that of Least Squares GAN.

$$\min_D V_{LSGAN}(D)$$
$$= \frac{1}{2}\mathbb{E}_{x,x_c\sim p_{data}(x,x_c)}[(D(x,x_c)-1)^2]$$
$$+ \frac{1}{2}\mathbb{E}_{z\sim p_z(z),x_c\sim p_{data}(x_c)}[D(G(z,x_c),x_c)^2]$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z\sim p_z(z),x_c\sim p_{data}(x_c)}[(D(G(z,x_c),x_c) - 1)^2]$$

The results show that SEGAN works well as an end-to-end method for speech enhancement. It was preferred over a baseline method most of the time when compared to the original noisy signal. An issue that was stated was the need to remove high frequency artifacts.

3.3 SpecGAN

A prominent method of applying audio data to GAN is to convert audio-forms into spectrograms (see Figure 1) using methods such as a short-time Fourier transform (STFT). Although commonly used, spectrograms can be non-invertible (due to lost phase information) and require an inversion method, such as that of least-squares error estimation (LSEE) [2]. SpecGAN uses a spectrogram model that works well with GAN algorithms already adapted to images. The audio is processed, normalized, clipped, and scaled to be represented as a spectrogram. The algorithm used to train is DCGAN (explained previously). Phase shuffle is introduced with the discriminator to help distinguish between artifact and non-artifact samples. Finally, the spectrogram is inverted, and the processing reversed to obtain an audio form [2]. SpecGAN was intended to serve as a baseline algorithm to compare to the WaveGAN algorithm. Both WaveGAN and SpecGAN were trained on speech command subset (SC09) of spoken positive integers 0-9.

3.4 WaveGAN

The more refined algorithm of WaveGAN is also introduced by [4] to serve as an improvement and comparison to SpecGAN. The WaveGAN architecture is also based off DCGAN. The WaveGAN parameters were adjusted to be able to be used with raw audio. Also, changes were made to have the same key features of DCGAN [4]. To do this, one dimensional filters with a length of 25 were used and an up-sample factor of 4 was used. An extra layer was added so that the audio length was approximately one second. Phase shuffle was also introduced to improve the output [2]. WGAN-GP was used to train the examples presented in the paper. The study used several datasets including drum sounds, bird vocalizations, TIMIT dataset, and classical piano pieces. The SC09 dataset was used for qualitative human grading with categories including accuracy, quality, ease, and diversity. Quantitative scoring was also done using inception scoring

and nearest neighbor comparisons. The results found that most of the time, listeners preferred WaveGAN (with phase shuffle of n=2) [2]. Benefits of the WaveGAN architecture are that it can operate on raw audio and is fully parallelizable. Although relatively new, WaveGAN has shown that future GAN research in audio generation can be used successfully in an unsupervised setting.

3.5 MuseGAN

The complexity of music has made it difficult to model and synthesize without human supervision. Musical pieces not only contain coherent hierarchical structures but multiple instrument that flow together. Music is based on many other factors, such as timing, rhythm, chord progressions, emotion, and flow. Symbolic music (such as MIDI) is a form of electronic music communication that is common in the audio world. In Multi-Track Sequential GAN (MuseGAN) [18], a new GAN architecture is proposed to generate polyphonic symbolic music. The study does so via three separate methods, although only the hybrid model will be mentioned (see Figure 3). The hybrid model is a combination of the other two models (composer and jamming model). The hybrid model consists of M generators which takes inter-track random vector $\mathbf{z}$ and intra-track random vector $\mathbf{z_i}$. Each generator takes an individual intra-track vector while they all get one inter-track vector. Only one discriminator is used to evaluate the generators. Figure 3 shows the basic structure of the hybrid model. The results of MuseGAN give a coherent generation of polyphonic music. The tracks generated follow the structure of human made tracks that can pass for enjoyable music (subjectively). Audio samples are provided in [18] which show promise in the field of symbolic music generation using GAN.

3.6 Applicable GAN Variants

There are some notable GAN variants that show promise and may be adapted to the previous works shown. The progressive growing of GANs [10] is a method used to slowly grow the generator and discriminator progressively together. The study uses images as a dataset and can achieve high resolution images from the CELEBA database. The study begins training with low resolution datasets and increases image resolution by adding layers to generate finer details. By focusing training on low resolution data, comparable results are obtained two to six times faster [10].

Another study tries to obtain the benefits from GAN without having to train a discriminator [19]. Generative Latent Optimization (GLO) shows promising results that have the benefits of GAN, such as image synthesis, sample interpolation, and linear arithmetic with noise vectors. GLO can also be used in a conditional setting. Further work needs to be done to achieve the same quality as current GAN architectures.

4. CONCLUSION AND FUTURE DIRECTION

In conclusion, this paper serves to provide an overview of the current state of audio enhancement and synthesis using different GAN architectures. While audio has not received as much attention compared with images, it is still a growing field with room to grow and improve. The papers surveyed provide a general introduction as to what has and can be done with GAN regarding audio. There is a large focus on speech enhancement and synthesis [11][12][13][3][14] which is applicable in the real world and widely used across different platforms. Music generation has many challenges to overcome due to the complexity that comes with music. Still, there has been progress with GAN that can be applicable in the sound effect and music industry [2][18]. Further work may require combining the best properties of various GAN architectures [5][8][10][15][9] to improve existing structures. Also, architectures with a combination of audio, images, and video may also serve a purpose and can be used to train each element jointly, [20] is a good example of combining video with audio. Overall, GAN has been a wide success since its introduction in 2014 and a plethora of research has been done. Many variants exist, and each can be useful in their own respect. Further research can still be done to perfect the algorithms available or even to introduce new algorithms.

5. ACKNOWLEDGMENTS

Special thanks to the department of Electrical and Computer Engineering at the University of Nevada, Las Vegas, and those who provided constructive feedback.

6. REFERENCES

[1] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in neural information processing systems, pp. 2672-2680. 2014.

[2] Donahue, Chris, Julian McAuley, and Miller Puckette. "Adversarial Audio Synthesis." arXiv preprint arXiv:1802.04208v2 (2018).

[3] Yang, Shan, Lei Xie, Xiao Chen, Xiaoyan Lou, Xuan Zhu, Dongyan Huang, and Haizhou Li. "Statistical parametric speech synthesis using generative adversarial networks under a multi-task learning framework." In 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 685-691. IEEE, 2017.

[4] Briot, Jean-Pierre, Gaëtan Hadjeres, and François Pachet. "Deep learning techniques for music generation-a survey." arXiv preprint arXiv:1709.01620 (2017).

[5] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." arXiv preprint arXiv:1701.07875 (2017).

[6] Creswell, Antonia, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. "Generative adversarial networks: An overview." IEEE Signal Processing Magazine 35, no. 1 (2018): 53-65.

[7] Wang, Kunfeng, Chao Gou, Yanjie Duan, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang. "Generative adversarial networks: introduction and outlook." IEEE/CAA Journal of Automatica Sinica 4, no. 4 (2017): 588-598.

[8] Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. "Improved training of wasserstein gans." In Advances in Neural Information Processing Systems, pp. 5767-5777. 2017.

[9] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

[10] Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

[11] Kaneko, Takuhiro, Hirokazu Kameoka, Nobukatsu Hojo, Yusuke Ijima, Kaoru Hiramatsu, and Kunio Kashino. "Generative adversarial network-based postfilter for statistical parametric speech synthesis." In Proc. ICASSP, vol. 2017, pp. 4910-4914. 2017.

[12] Michelsanti, Daniel, and Zheng-Hua Tan. "Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification." arXiv preprint arXiv:1709.01703 (2017).

[13] Pandey, Ashutosh, and Deliang Wang. "On adversarial training and loss functions for speech enhancement." In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5414-5418. IEEE, 2018.

[14] Yeh, Cheng-chieh, Po-chun Hsu, Ju-chieh Chou, Hung-yi Lee, and Lin-shan Lee. "Rhythm-Flexible Voice Conversion without Parallel Data Using Cycle-GAN over Phoneme Posteriorgram Sequences." arXiv preprint arXiv:1808.03113 (2018).

[15] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784 (2014).

[16] Zhao, Yi, Shinji Takaki, Hieu-Thi Luong, Junichi Yamagishi, Daisuke Saito, and Nobuaki Minematsu. "Wasserstein GAN and Waveform Loss-based Acoustic Model Training for Multi-speaker Text-to-Speech Synthesis Systems Using a WaveNet Vocoder." IEEE Access (2018).

[17] Pascual, Santiago, Antonio Bonafonte, and Joan Serra. "SEGAN: Speech enhancement generative adversarial network." arXiv preprint arXiv:1703.09452 (2017).

[18] Dong, Hao-Wen, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. "MuseGAN: Symbolic-domain music generation and accompaniment with multi-track sequential generative adversarial networks." arXiv preprint arXiv:1709.06298 (2017).

[19] Bojanowski, Piotr, Armand Joulin, David Lopez-Paz, and Arthur Szlam. "Optimizing the latent space of generative networks." arXiv preprint arXiv:1707.05776 (2017).

[20] Owens, Andrew, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H. Adelson, and William T. Freeman. "Visually indicated sounds." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2405-2413. 2016.