Conditional Generative Adversarial Network with One-Dimensional Self-Attention for Speech Synthesis

Nirav Jain[1], Yash Javeri[1], Sudhir Bagul[1]

[1] Computer Department, Dwarkadas J. Sanghvi College Of Engineering, Vile Parle, Mumbai, India
niravjain1709@gmail.com, yjaveri99@gmail.com, sudhir.bagul@djsce.ac.in

Abstract. Previous works like [1] have shown that it is possible to reliably generate coherent waveforms of good quality by employing a Generative Adversarial Network(GAN). With an intent to improve the stability of the GAN and enhance the quality of the generated audio, we propose a modified version of MelGAN. We combine Self-Attention with MelGAN architecture. Self-attention GAN [2] have shown improved results in the generation of high-quality images. To make use of the Self-Attention layer's quality of establishing long-range dependencies in audio generation[2], we embed a one-dimensional self-attention within the MelGAN's generator. In this paper, we use the LJ Speech Dataset to train MelGAN as well as our proposed architecture. To interpret the results, we calculate and compare the Mean Opinion Scores and Mean Opinion Score- Listening Quality Objective of both the architectures trained until the thousandth epoch.

Keywords: Deep Learning, Generative Models, Audio Generation, Audio Processing, Convolutional Neural Networks, Intelligent Systems

1 Introduction

Using computers to generate natural-sounding audio has been a research topic for quite a while now, but no satisfactory solution has been found yet. This research has applications in a wide variety of fields like entertainment, hospitals, etc. However, generating human-like audio is not an easy task because audio is a complex data format. It involves several intricacies like frequency and amplitude which govern different aspects of an audio clip like pitch, loudness, etc. Also, the high temporal resolutions of an audio ranging from 16 kHz to 192 kHz make the task even more difficult. Furthermore, the network must understand and establish meaningful long-term and short-term dependencies that exist in an audio file.

Keeping in mind the challenges involved in directly generating raw audios, most approaches usually model a low-resolution representation like mel-spectrogram. Even in the case of text to audio, the process is generally divided into two parts, converting text to an intermediate form and converting this intermediate form to raw audio. Various methods have been tried to produce raw audio from intermediate forms like mel spectrograms or aligned linguistic features. The most generic way of producing raw audio from such formats is by using signal processing methods like Griffin-Lim [5]. However, the problem with such methods is that they introduce a lot of noise, disturbances and unclear sound. Another approach would be to use neural networks for the same. Models like WaveNet[3] produce state-of-the-art results but at the cost of slow inference speed. Similarly, parallel WaveNet [14] produces good results at the cost of high memory. Such solutions cannot be used in real-time applications and hence need improvement. One model that introduces such improvements is MelGan [1] which is lightweight and easy to train. It uses GAN's to achieve it's high and reliable performance. The use of GAN's to generate audio has never been explored before effectively but the performance of GAN's in producing high-quality images motivates the research in this particular area.

The proposed paper focuses on improving the performance of MelGAN by introducing Self-Attention [2] in the MelGAN architecture. Self-attention has been used in GANs before, to generate images, and the addition of the Self-Attention layer has been shown to improve the quality of images to a great extent. We try to achieve similar results for 1-Dimensional audio data by adding a self-attention layer in the generator

of MelGAN. Self-attention in [2] was used to establish long-range dependencies within an image. Such dependencies are present in a much more considerable amount in an audio file because human language is structured in such a way that each part of a sentence is related to some other part of the same sentence. Hence establishing such dependencies is an important task in the audio generation and can be easily achieved using a 1-Dimensional self-attention layer. We train the proposed network on LJ-Speech Dataset for 1000 epochs and then compare the results obtained with MelGAN.

2 Literature Review

In [1] "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis," the paper proposes to generate raw audio waveforms from Mel-spectrograms by training GANs. Mel-spectrogram is given as an input to the generator, a fully convolutional feed-forward network which generates the output. The generator consists of upsampling layers followed by residual blocks. These upsampling layers with transposed convolutional layers help in upsampling the input by 256x. The researchers adopt a multiscale architecture with three discriminators, each operating on different scales of audio. In order to test the significance of the various architectural decisions, the model is trained and unit-tested by removing the various key parts of the proposed architecture one-by-one. The observations show that dilated convolutional stacks, weight normalization and the multi-scale discriminator architecture(instead of single discriminator) play a vital role in the performance of MelGAN. MelGAN can potentially be a plug-and-play replacement in various higher-level audio generation tasks like the [7] Universal Music Translation Network or the [8] Vector-Quantized VAEs.

In [2], the authors propose Self-Attention GAN's to generate high-quality images. Normal GAN's cannot establish long-range dependencies as they focus on only spatially local points of an image. The paper tries to overcome this limitation by mapping long-range dependencies in an image by using attention maps. In self-attention, the value at any position considers the features at all the positions and hence, provides a better representation of the overall image. The paper proposes to use self-attention in both the generator as well as the discriminator. The paper also uses spectral normalization in both the discriminator and the generator and helps improve the training dynamics by stabilizing the generator. It assists in preventing unusual gradients in the generator. The paper shows that the trained model performed best when the attention mechanism was applied to larger feature maps compared to smaller ones. It becomes easier to map dependencies as it has more evidence to choose from the image. The model shows promising results compared to previous state-of-the-art models with an Inception score of 52.52 and a Frechet Inception distance of 18.65.

The authors of the paper [3] "Wavenet: A generative model for raw audio" propose a probabilistic and autoregressive model whose output depends not only on the current input but also on all the previous inputs. The model takes a Mel spectrogram as input and provides the corresponding audio for each time step as the output. The most crucial aspect of Wavenet, as proposed by the authors, is the dilated causal convolutions, which help to correctly predict the audio at any given time steps without considering Mel spectrograms from any future time steps. The problem with causal convolutions is that they have a small receptive field. To overcome this obstacle, the authors propose using stacked dilated convolutions to increase the receptive field by orders of magnitude with just a few layers. The paper also proposes the use of softmax distributions with non-linear quantization for better results. To further improve the results and training speed, the paper proposes gated activation units and residual connections. The model was used for three tasks: multi-speaker speech generation, text to speech, and music audio modeling. In terms of the naturality of audio, the model outperforms all its previous methods of text to speech generation and shows promising results in the other two tasks as well.

In [4], the researchers propose a flow-based network to generate audios from mel-spectrograms by combining ideas from Glow[4] and Wavenet[3]. Unlike [4], the paper posits and shows that the autoregressive model is unnecessary for synthesizing speech. Waveglow is a generative model that samples a simple input distribution. The authors have used zero mean spherical Gaussian distribution with dimensions equal to the desired output's dimensions. In order to make the training stable and more straightforward, the model uses only a single network with one cost function that is "maximizing the negative likelihood of the training data." Waveglow uses twelve coupling layers and twelve invertible convolution layers, with each having eight dilated convolution layers. Only with slight differences,

Waveglow has the maximum Mean Opinion score amongst other speech-synthesis architectures like wavenet[3], Griffin-Lim[5]. In terms of speed of inference speed, even the unoptimized implementation of WaveGlow is slightly faster than Wavenet with a synthesis rate of 500kHz on an NVIDIA V100 GPU. The model has an advantage over others with respect to the speed of inference and simplicity of training.

3 Proposed Methodology

Fig. 1 illustrates the proposed architecture of the generator of SA-MelGAN, designed by combining ideas from MelGAN and SAGAN, following the general adversarial game between the generator and the discriminator. As shown, we introduce a self-attention block in the model architecture in order to improve its performance. The MelSpectrogram is given input to the first one-dimensional convolution layer with a kernel of size 7. This output then undergoes a total of 256x upsampling in the consecutive blocks. The upsampling is done in four stages 8x, 8x, 2x, 2x by adjusting the stride values. Each upsampling layer consists of a one-dimensional transposed convolution layer with stride values depending upon the stage of upsampling and the kernel sizes equal to twice that of the stride. LeakyReLU has been used as the activation function in all the upsampling blocks. Each upsampling output is given to a residual stack block containing dilated convolutional layers. All the levels of the generator as well as the discriminator use 1D Reflection padding to ensure that the outputs are of the required size.

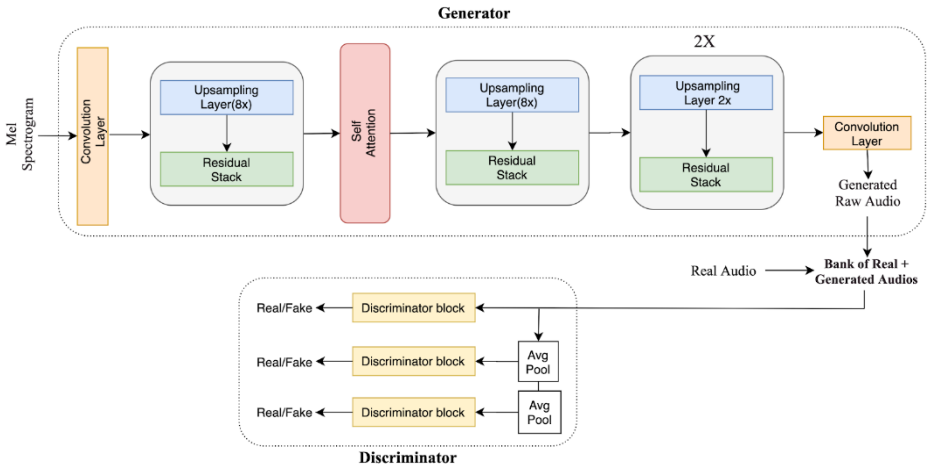

Fig. 1. Proposed SA-MelGAN Architecture

3.1 Residual Stack

Each block consists of 3 dilated convolutional layers with dilation values equal 3, 6 and 9, respectively, for the three layers. The dilated convolution later is followed by a regular convolutional layer with a kernel size of 1. This output is coupled with the initial input passing through a bunch of convolution layers. Dilated convolutions help increase the receptive field significantly without any extra computation power or loss of resolution.

3.2 Self-Attention

The self-attention layer, as shown in Figure 1, is added after the first residual block. We have used 1-D self-attention, which takes an input having 256 channels and 80x1 dimensions obtained from the first residual stack. This input is first passed through three 1x1 convolutions, as shown in Fig 2., to reduce the dimensionality of the input received from the previous layer.

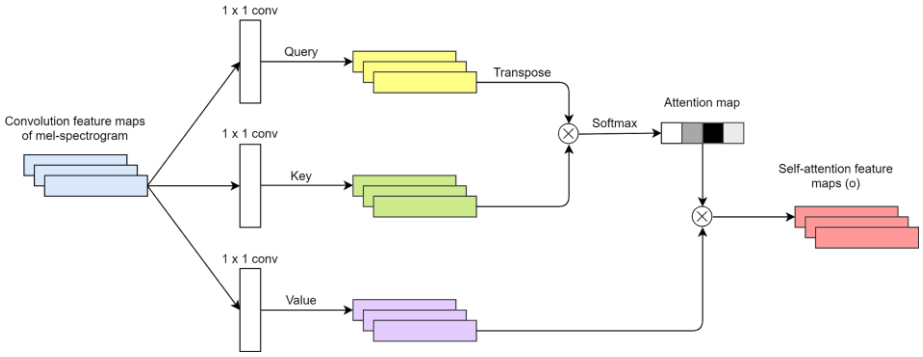

Fig. 2. One-Dimensional Self-Attention

Since the data is 1-Dimensional, we use 1D convolutions with kernel size 1 to achieve the same effect that a 2D 1x1 convolution has. More precisely,

$$\text{Query} = \text{f(x)} = \text{W}_f\text{x}$$
$$\text{Key} = \text{g(x)} = \text{W}_g\text{x}$$
$$\text{Value} = \text{h(x)} = \text{W}_h\text{x}$$
$$Where, x \in R^{CxN}, W_f \in R^{\underline{C}xC}, W_g \in R^{\underline{C}xC}, W_h \in R^{\underline{C}xC}$$
$$\underline{C} = C/8N \text{ is the dimension of the previous layer}$$
$$i.e. N = Lx1 \text{ where L is the length of the } 1-D \text{ array}$$

Now, we have three outputs which are used to calculate the self-attention viz. query, key, and value. First, we transpose the query and matrix-multiply it with the key. Then we pass this multiplied matrix through a softmax function to obtain the attention map.

$$s_{ij} = f(x_i)^T g(x_j)$$
$$\beta_{j,i} = \frac{exp(s_{ij})}{\sum_{i=1}^{N} exp(s_{ij})}$$

This attention map is then matrix-multiplied with the value matrix to obtain the self-attention feature maps. Furthermore, this output is multiplied with a scalar (which is learned during training) and added back to the input. These feature maps are then used by the network to reconstruct the audio.

$$o_j = \sum_{i=1}^{N} \beta_{j,i} h(x_i), o \epsilon R^{CxN}$$
$$y_i = \gamma o_i + x_i$$

All the convolutions used to calculate self-attention in our implementation are 1D because of the nature of the input from the previous layers. Also, the Self-attention layer is located at the beginning to efficiently map the long-range relationships in the input Mel-spectrogram in a memory-efficient way. For example, if mel-spectrogram of dimension (80,150) is given as the input, the input dimension that the self-attention layer in the proposed model gets is (16,256,1200) (the format used is (batch_size, channels, dimensions)), which then converts to 3 matrices of size (16,32,1200) when we apply 1x1 convolutions. These three matrices are the query, key, and value matrices. After transposing the query and matrix multiplying it with the key, the size of the input (150) increases by eight times (1200), resulting in a matrix of the dimension (16,1200,1200), which corresponds to utilization of 171.66 MB of the GPU. On the other hand, if self-attention is embedded after the second residual stack, after matrix multiplication of query with the key, the input size increases by 64 times, resulting in a matrix of dimension (16,9600,9600). This corresponds to 10.98GB of GPU memory. Hence, the self-attention layer's proposed location is after the first residual stack where matrix multiplications for calculating attention can be carried out more efficiently.

3.3 Normalization Technique

Which normalization technique to use is a vital decision. Image generation GAN architectures like [9] use instance normalization. However, according to [1], instance-normalization removes essential information related to the pitch and tends to generate metallic audios. Spectral normalization also generates poor results, which affects the generator's feature mapping objective. Weight normalization works best in this case as it does not mitigate the discriminator's capacity in any way as it simply reparameterizes the weight matrices. Therefore, we use the weight normalization technique for the discriminator as well as for the generator.

3.4 Dataset and Preprocessing

We have used the LJ Speech Dataset [12], consisting of 13100 audio clips, each varying in length from 1 to 10 seconds. All the clips are read by a single speaker from 7 non-fiction books. We use a mel-spectrogram as an input to our model. It is a representation in which the audio signal in the time domain is mapped onto the mel-scale. First, the audio is converted into a frequency domain, and then these frequencies are mapped to the mel-scale using a non-linear transformation. The crucial information from audio, such as its loudness and amplitude, varies over time at different frequencies and can be efficiently obtained from a mel spectrogram. To convert the audio into a mel-spectrogram, we have used a window length of 1024 and a hop length of 256, according to which the audio is sampled for computing the Fast Fourier Transform. The frequency spectrum is then divided into 80 mel channels that are then mapped to the mel scale to obtain the mel spectrogram. The proposed methodology uses audios with a sampling rate of 22050Hz. In order to have all the audios of the same length while training, we pad the wav file with zeros. Also, the frequencies of the generated mel-spectrogram are restricted between 0Hz and 8000Hz. We divide the dataset into a 4:1 ratio for training and validation, respectively. Hence, out of 13100 audio-mel pairs, 10480 instances are used for training, while the remaining 2620 are used for validation.

3.5 Training

We use the hinge loss version of the GAN objective [10] for training.

$$min_{D_k}\, E_x[min(0,1 - D_k(x))] + E_{s,z}[min(0,1 + D_k(G(s,z)))], \forall k = 1,2,3$$

$$min_G\, E_{s,z}\left[\sum_{k=1,2,3} -D_k(G(s,z))\right]$$

where x denotes the raw waveform, s denotes the conditioning information and z represents the Gaussian noise vector.

To minimize the L1 distance between the discriminator feature maps of the real and synthetic audio, we also use feature matching objectives [10] for the generator's training along with the discriminator's signal. L1 loss is not added in the audio space. This is because it introduces noise in the audio and affects its quality as mentioned in the work [1].

$$L_{FM}(G, D_k) = E_{x,s\sim p_{data}}\left[\sum_{i=1}^{T}\frac{1}{N_i}|| D_k^{(i)}(x) - D_k^{(i)}(G(s)) ||_1\right]$$

where Dk(i) denotes the ith layer's feature map output of the kth discriminator block, Ni represents the number of units in each layer. We use this feature mapping at all the intermediate layers of each discriminator block.

In conclusion, to train the generator, we use the following objective function.

$\lambda = 10$ as in [11]

$$min_G \left(E_{s,z} \left[\sum_{k=1,2,3} -D_k(G(s,z)) \right] + \lambda \sum_{k=1}^{3} L_{FM}(G, D_k) \right)$$

Both the architectures are trained with a batch size of 16 on google colab, which provides an Nvidia K80 GPU with 12 GB of memory, 0.82 GHz clock speed, and 4.1 TFlops of performance. We use 'adam' as the optimizer with a learning rate of 0.001, $\beta 1$ as 0.5, and $\beta 2$ as 0.9.

4 Results

We carry out qualitative and quantitative analysis between our proposed architecture and MelGAN. From figures 3 and 5, it can be seen that as the discriminator loss decreases i.e. as it becomes more accurate in distinguishing real and generated audio, the generator loss increases (figures 4 and 6), forcing the generator to produce even better results. This indicates that the architectures are correctly implemented, and the GANs are getting trained on the right track. Observing the change in loss values, as the models train progressively, it is seen that MelGAN's loss values oscillate relatively more as compared to that of SA-MelGAN. From these observations, it can be concluded that SA-MelGAN trains more stably than MelGAN.

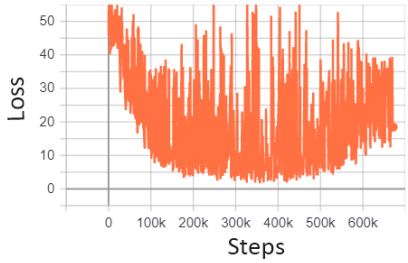

Fig. 3. Discriminator loss for MelGAN

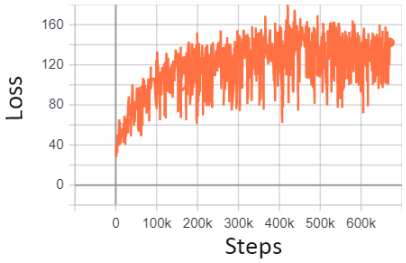

Fig. 4. Generator loss for MelGAN

In the case of MelGAN, at around 545,100th step, the loss of the discriminator while training again starts increasing. This gradual increase in discriminator loss leads to some fluctuations in the generator as well. This increases the time taken by the GAN to reach the state of equilibrium. On the contrary, such changes are not seen in SA-Melgan (figures 5).

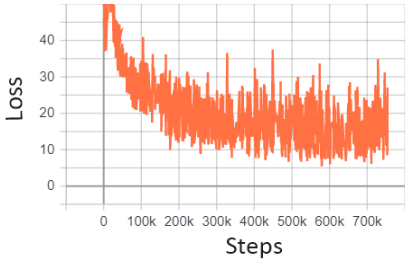

Fig. 5. Discriminator loss for SA-MelGAN

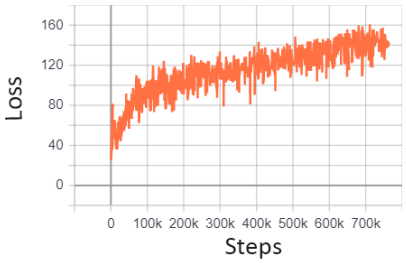

Fig. 6. Generator loss for SA-MelGAN

For the qualitative analysis, we randomly select five audio samples from each category viz. the reconstructed audios by MelGAN, reconstructed audios by SA-MelGAN, and original raw audios. These 15 audio clips are mixed and rearranged. We then ask 120 raters to rate these audio clips on a scale of 1 to 5, keeping the label (original or generated) of the audio hidden. In order to calculate the Mean Opinion Score, we average the ratings within each category. The MOS for every category is mentioned in table 1. Table 1 shows that the MOS for SA-Melgan is greater than Melgan by almost 8.94%. While few of the audios generated by SA-MelGAN sound less naturalistic than those generated by MelGAN, most of them generated by SA-MelGAN seem very close to the original ones.

We also calculate a MOS-LQO (Mean Opinion Score- Listening Quality Objective) score that ranges from 1 (worst) to 5 (best). It is an objective and full-reference metric for perceived audio qualities. We calculate this metric using the ViSQOL i.e. Virtual Speech Quality Objective Listener [15], an open-source tool by Google. The main idea behind ViSQOL is that it calculates the spectro-temporal measure of similarity between the real and generated audios. Table 1 demonstrates the results received by averaging the MOS-LQO values of 20 random audio samples each generated by both architectures. Although SA-MelGAN's MOS (human evaluation) was 0.3 greater than that of MelGAN, we did not observe a significant increase in the computer-generated MOS-LQO.

Table 1. Mean Opinion Score (Human Evaluation) and Mean Opinion Score- Listening Quality Objective (MOS-LQO)

Category	Mean Opinion Score	MOS-LQO
Original	4.502	4.732
SA-MelGAN	3.646	3.298
MelGAN	3.346	3.207

Observing figures 7 and 8, it can be posited that the amount of overall noise present in the audio generated by SA-MelGAN is not significantly different from that of audio generated by MelGAN architecture. However, the maximum of all the noise values throughout the audio in the case of SA-MelGAN is less than that in MelGAN generated audios. Hence, the noise is not much noticeable to human ears in the case of SA-MelGAN. This attributes to the reason behind our better Human Evaluated MOS compared to the MOS of MelGAN and also the reason behind the insignificant difference in MOS-LQO between the two architectures.

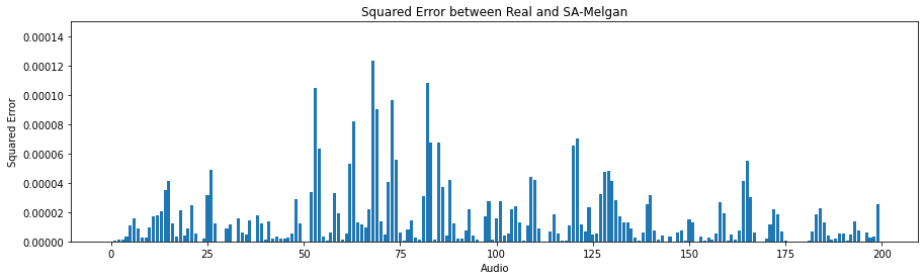

Fig. 7. Squared Error plot for SA-MelGAN

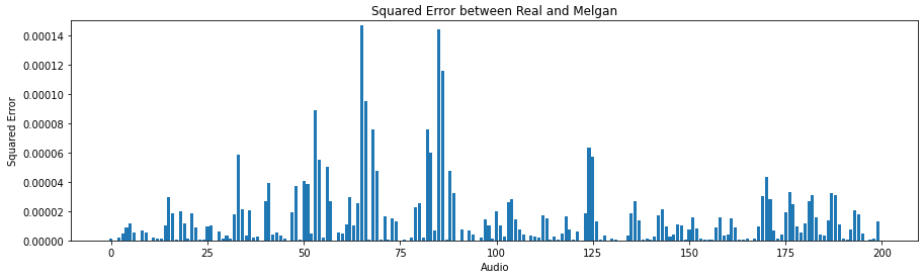

Fig. 8. Squared Error plot for MelGAN

5 Conclusion

In this paper, we have introduced a modification to MelGAN. We have added a Self-attention layer after the first residual stack and we see an improvement in the Mean Opinion Score thus obtained compared to MelGAN. Although there is a notable increase in the Mean Opinion Score, we observe only a slight increase in MOS-LQO. Also, the higher peaks in squared error in the case of MelGAN are the reasons for its low Human-Evaluated MOS but almost equal MOS-LQO. Hence, we conclude that the addition of a self-attention layer for audio generation in Melgan improves its overall quality with minor errors. However, Melgan produces audio with higher and noticeable fluctuations without this layer when trained till the 1000th epoch. Self-attention also facilitates a smooth and stable convergence with the addition of only a few parameters. Although there is a negligible difference in the inference time between the proposed model and Melgan, it should be noted that the proposed model uses memory intensive calculations. Hence further research is required in this field to reduce the memory requirements and improve the quality of the audio produced.

References

1. Kumar, K., Rithesh Kumar, T. D. Boissiere, L. Gestin, Wei Zhen Teoh, J. Sotelo, A. D. Brébisson, Yoshua Bengio and Aaron C. Courville. “MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis.” ArXiv abs/1910.06711 (2019): n. pag.
2. Zhang, Han, Ian J. Goodfellow, D. Metaxas and Augustus Odena. “Self-Attention Generative Adversarial Networks.” ArXiv abs/1805.08318 (2019): n. pag.
3. Oord, A., S. Dieleman, H. Zen, K. Simonyan, Oriol Vinyals, A. Graves, Nal Kalchbrenner, A. Senior and K. Kavukcuoglu. “WaveNet: A Generative Model for Raw Audio.” ArXiv abs/1609.03499 (2016)
4. Prenger, Ryan, Rafael Valle and Bryan Catanzaro. “Waveglow: A Flow-based Generative Network for Speech Synthesis.” ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2019): 3617-3621.
5. Kingma, Diederik P. and Prafulla Dhariwal. “Glow: Generative Flow with Invertible 1x1 Convolutions.” ArXiv abs/1807.03039 (2018)
6. Griffin, D. and J. Lim. “Signal estimation from modified short-time Fourier transform.” IEEE Transactions on Acoustics, Speech, and Signal Processing 32 (1984): 236-243.
7. Oord, A., Oriol Vinyals and K. Kavukcuoglu. “Neural Discrete Representation Learning.” NIPS (2017).
8. Mor, Noam, Lior Wolf, A. Polyak and Yaniv Taigman. “Autoencoder-based Music Translation.” (2018).
9. Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou and Alexei A. Efros. “Image-to-Image Translation with Conditional Adversarial Networks.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): 5967-5976.
10. Larsen, Anders Boesen Lindbo, Søren Kaae Sønderby, H. Larochelle and O. Winther. “Autoencoding beyond pixels using a learned similarity metric.” ArXiv abs/1512.09300 (2016): n. pag.
11. Wang, T., Ming-Yu Liu, Jun-Yan Zhu, A. Tao, J. Kautz and Bryan Catanzaro. “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs.” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018): 8798-8807.
12. Keith Ito and Linda Johnson. "The LJ Speech Dataset." https://keithito.com/LJ-Speech-Dataset/. (2017).
13. Oord, A., Y. Li, I. Babuschkin, K. Simonyan, Oriol Vinyals, K. Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, S. Dieleman, E. Elsen, Nal Kalchbrenner, H. Zen, A. Graves, Helen King, Tom Walters, D. Belov and Demis Hassabis. “Parallel WaveNet: Fast High-Fidelity Speech Synthesis.” ArXiv abs/1711.10433 (2018): n. pag.
14. Chinen, M., Felicia S. C. Lim, J. Skoglund, Nikita Gureev, Feargus O'Gorman and A. Hines. “ViSQOL v3: An Open Source Production Ready Objective Speech and Audio Metric.” 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX) (2020): 1-6.