Proceedings of 12th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2022 Month: October Volume: 12

Generative Adversarial Network for Audio Compression

Bishal Chaulagain [a], Shikhar Bhattarai [b]

[a, b] *Department of Electronics and Computer Engineering, Thapathli Campus, IOE, Tribhuvan University, Nepal*
✉ [a] bishal76msiise05@tcioe.edu.np, [b] shikhar28@tcioe.edu.np

Abstract
There has been a massive increase in the volume of data produced. New sorts of data are being created, such as genomic data, VR data, 3D data, 360-degree autonomous driving data, and cloud data. In order to create excellent compressors, a lot of human effort is put into examining the statistics of these new data formats. Signal compression is a useful technique for lowering transmission expenses and extending the life of the signal produced. To simplify signal compression an audio compression system based on a generative adversarial network (GAN) is presented here. The audio signal is processed to convert into the frequency domain and the audio signal spectrum is converted into Mel-spectrogram which is fed into an encoder that produces the latent vector. The latent vector representing the compressed signal is supplied into a generator network that has been trained to create high-quality signals that minimize the target of the objective function. Non-uniformly quantized optimum latent vectors are discovered by back-propagation using the optimization method iteratively to efficiently quantize the compressed signal. Subjective and objective evaluations including PESQ and MUSHRA are used to evaluate the Proposed signal compression method's performance compared with BPGAN, CELP and Opus techniques.

Keywords
ADMM, Back Propagation, GAN, Latent Vector Encoding, Signal Compression

1. Introduction

The audio quality is continually increasing in today's data streams due to sophisticated devices used which produce high-quality signals that demand higher storage capacity and higher communication bandwidth. It is impractical to store and communicate this large volume of data generated in the information-rich world, and therefore signal compression is essential to manage those signals with limited communication bandwidth and storage space, which seems to be a substantial challenge to practical application. For the purpose of creating effective compressors, a lot of human effort is put into evaluating the statistics of these new data formats. An essential instrument for cutting communication expenses and extending the life of the signal generated is signal compression. A generative adversarial network (GAN)-based audio compression system was presented to aid in signal compression.

The amount of data created has dramatically increased in the big data era. Genomics data, virtual reality data, 3D data, 360-degree autonomous driving data, and cloud data are a few of the new types of data that are being produced. A lot of human work is invested into analyzing the statistics of these new data formats in order to develop excellent compressors. To reduce the network bandwidth utilization on the communication channel shared by numerous edge devices, signal compression is a task of critical importance. Additionally, signal compression greatly lowers the energy overhead associated with wireless data transmission, which frequently accounts for the majority of the energy used by power constrained IoT devices. Inspired by the recent spectacular success of generative adversarial networks (GAN) in numerous applications, a GAN-based compression framework known as backpropagated GAN was implemented with a novel optimization method, where the compressed signal is represented by a latent vector fed into a generator network that has been trained to produce realistic high-quality signals.The signal compression finds applications in the bandwidth and storage-constraint IoT devices and networks. Other applications include signal broadcasting like TV, Radio, and FM broadcasting along with the fixed and mobile communication networks to reduce bandwidth consumption.

2. Literature Review

Due to the outstanding performance of GAN attained in the various image processing application including generation, synthesis, compression, enhancement, and translation [1] it was introduced to the audio sector for audio enhancement and compression which consequences in an impressive result opening the field of research on audio using GAN. Han et al on [2] first proposed the non-uniform quantization in a Deep neural network using the K-means algorithm for deep weight compression, where the centroid of the cluster was updated during the training phase that intensely reduced the bit length of Deep Neural Network Weight. To overcome unnecessary overhead for compression Leng et al on [3] develop extreme image compression for encoder-decoder style networks inserting the differentiable quantization module.

The Deep Neural Network and autoencoder-based image compression developed by Balle et al on [4] and Rippel et al on [5] was a revolutionary achievement in the field. Where [4] focuses on MSSSIM multiscale structural similarity for the assessment of quality image between original and decompressed one and MSE optimizing. Theis et al on [6] proposed a system that compresses the image by applying the method using traditional quantization and encoder for the reduction of bitrate. Recent models adopt GAN-based encoder for image compression such as Mentzer et al on [7], Agustsson et al [8] achieves a high compression rate with visually attractive image. Even though the details provided by those decoders seem to distort the actual image details. Recent model trends that were deployed by Minnen et al on [9], Lee et al on [10], and Cheng et al on [11] use context-adaptive entropy-based image compression, these make the use of additional bits allocated by hyperpriors for bit consuming and complex context whereas for easily inferred contexts, autoregressive models are used.

Initially, the traditional audio codecs like CELP [12], totally open and royalty-free Opus [13], and (AMRWB) Adaptive multi-rate wideband [14] are generally used for simple data processing. This processing is mainly based on the features that are constructed for acceptable audio quality demand for a higher bit rate greater than 16 kbps.

The DNN based approach employed by Kankanahalli et al on [15] confirmed the possibility of end-to-end training of the audio codec that shows similar performance to that of traditional handcrafted AMRWB codec at 9-24 kbps. Besides this Cernak et al on [16] employing the deep spiking neural network SNNs with synthesizers and paired phonological analyzer shows 369 bps of bitrate audio codec keeping only speaker identifier and content information to achieve the lower bitrate of 369 bps.

Another milestone for the synthesis of audio with higher quality is adopting the high-end audio codec realization like Wavenet by Oord et al on [17] and Wave RNN by Kalchbrenner on [18] using fine-tuned deep neural network DNN-based vocoders. that are considered as high-quality voice. Codecs such as [19], learned Wavenet was used to generate audio as an encoder that has the audio quality equivalent to that produced from the AMRWB. Besides this, Van et al on[20] proves that using (VQ-VAE) vector-quantized variational autoencoder framework the Wavenet vocoder is capable of generating quality audio by providing the discrete latent representation of audio. Also Garbacea et al on [21] contributes to reaching the bitrate up to 1.6 kbps but the system used here cannot address the low bitrate.

An important milestone was added by deploying the GAN for audio processing where Pascual et al on [22] use the SEGAN that established GAN can be used for the audio processing and shows the improved result on the audio processing task besides image processing. Further Donahue et al on [23], Marafioti et al on [24], and Engel et al on [25] claims that GAN can be used for simple speech and instrumental audio signal synthesis. However, generating an audio signal of high quality with a random hidden input signal is still a challenge. Hence for an audio compression proposed by Liu et al on [26] introduce the BPGAN that overcomes the limitations of previous work by achieving considerably the same quality for a lower bit rate by employing a combination of BPGAN along with Huffman encoder.

3. Methodology

3.1 Theoretical Formulations

Due to the remarkable inspirational succession of generative adversarial networks (GAN) on various applications, utilizing the same compression framework for audio compression seems interesting, and it canbe assumed that this could perform better. BPGAN which was originally proposed for the unified signal compression on [26] could be extended for the focused audio signal compression with advanced optimization technique and arithmetic encoder implementation. The audio signal in generative models is represented as a latent vector that was compressed by using iterative backpropagation to search for the optimal latent vector. The core idea is to compress the signal in the generative GAN in the form of a latent vector to search for the optimal latent vector through the iterative backpropagation at encoding to compress the target signal.

3.2 BPGAN Compression Model

The BPGAN compression framework may be used for any kind of signal type as long as possible to train the GAN model to produce a realistic type of output. The input signal x is encoded as initialized of the compressed signal $z_0 = E(x)$ at the initial stage after that z which is the latent vector was optimized and updated so as to minimize the objective function $F()$ through iteratively backpropagating from the generator $G()$. Here, the objective was the similarity measure of the input target signal x and reconstructed signal $G(z)$.

The optimal latent variable z is made discrete by applying the quantization scheme $Q()$, at the time of the continuous backpropagation process. The compressed signal is entropy encoded using Huffman encoder or arithmetic encoder before transmitting to the receiver end for further size reduction. The framework saves the generator parameter of the transmitter or compressor side and shares it between receiver and transmitter.

Considering the receiver side, for decoding the latent signal $G(\hat{z})$, the same generator parameter which was saved before was used which reconstructs the signal back to nearly original format through post-processing.

BPGAN compression is different than another GAN-based compression approach which relies only on an encoder for signal compression in the sense that the former performs the compression by iteratively updating and searching latent vector on generator input using the backpropagation that minimizes the objective function from generator output. The encoder here serves to accelerate the backpropagation by reducing the number of iterations, which ultimately improves the compression ratio and quality of the signal.

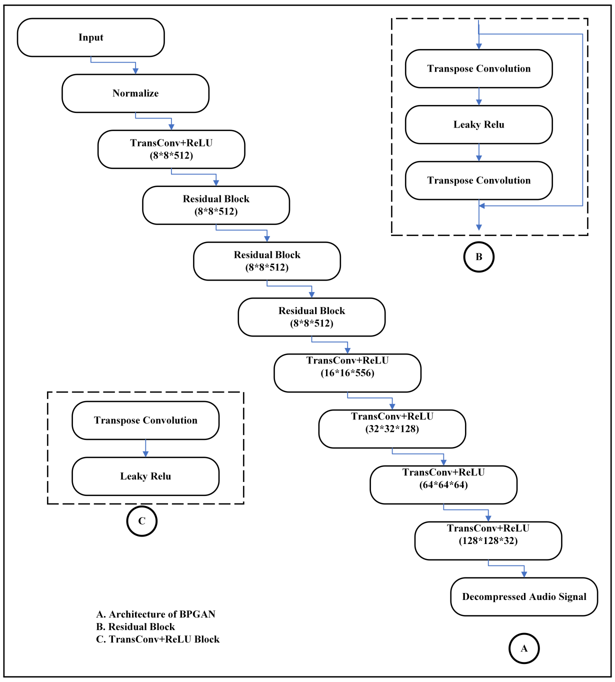


Figure 1: BPGAN Generator Network for Audio compression

3.3 System Block Diagram

The model for audio signal compression using GAN consists of preprocessing, Encoding, latent vector generation, Generator, loss function generation, Backpropagation, and Entropy encoder in the compression or transmitter block. The output of this transmitter block is the entropy-coded latent vector. Similarly, the decompression or receiver block consists of the entropy decoder, generator, and post-processing block. The output of this block is trying to reconstruct the original input signal as shown in figure below.

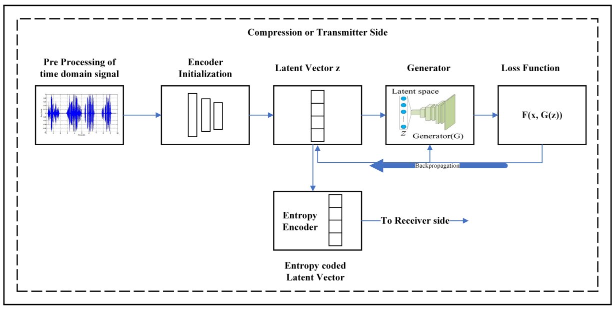


Figure 2: Block Diagram of the Proposed system transmitter side

The first block of the system is preprocessing block where the data is preprocessed to obtain the signal that is suitable for further processing. Here, mainly the time domain audio data set is converted into the spectral domain using STFT (short-time Fourier transform) as a result the data is converted into a spectrogram.

The next block consists of the encoder block where the input signal in the form of mel spectrogram is encoded as initialization of compressed signal, as a result, the signal is converted into latent vector z which is the third block of the system block diagram. The encoder is cascaded to the generator of the GAN to form the auto-encoder which is trained and maps from signal space to latent space.

The next block is the generator block of the GAN network. The generator and the discriminator of GAN produce the loss function which is the feedback to the generator and the latent vector, which in turn, updates and optimize the latent vector z to minimize the specific objective function. During this iterative backpropagation process, the optimal latent vector is discretized and quantized for the compression of the signal.

At the last stage, the latent vector is entropy encoded using arithmetic encoder for further signal compression and transmitting the signal to the receiver end.

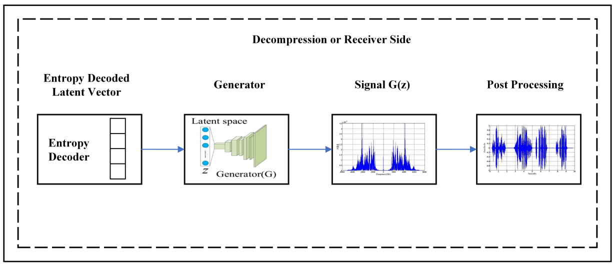


Figure 3: Block Diagram of the Proposed system receiver side

At the receiver, the signal is first entropy decoded using arithmetic decoder and generated signal using the same generator parameter which is trained at the transmitter side that produces the mel-spectrogram which is post-processed on post-processing block to convert into the audio signal using inverse STFT.

3.4 Dataset Explanation

The TIMIT corpus of read speech was created with the intention of providing speech data for acoustic-phonetic research as well as for the development and evaluation of automatic speech recognition systems. 6300 words altogether, each recorded at a sample rate of 16 kHz, were given by 630 speakers from 8 major American dialect regions for TIMIT. Texas Instruments, Inc., SRI International, and the Massachusetts Institute of Technology (MIT) worked on the corpus design. The speech was recorded at Texas Instruments, typed out at Massachusetts Institute of Technology, verified at the National Institute of Standards and Technology, and then prepared for CD-ROM production. Utilizing the TIMIT dataset, the voice compression network is trained.

Table 1: Dialect distribution of speakers

Dialect Regions	Male%	Female%	Overall%
dr1:	63	27	8
dr2:	70	30	16
dr3:	67	23	16
dr4:	69	31	16
dr5:	63	37	16
dr6:	65	35	7
dr7:	74	26	16
dr8:	67	33	5
Total (8)	70	30	100

3.5 Description of Algorithms

3.5.1 BPGAN Algorithm

```
Require: well trained generator $G(\cdot)$, encoder $E(\cdot)$
    pre-defined quantization function $Q(\cdot)$
    signal to be compressed $\boldsymbol{x}$
    objective function $F(\cdot)$
    quantized set $S$
Ensure: latent vector quantization $\tilde{\boldsymbol{z}}$
latent vector initialization $\boldsymbol{z}_0 = E(\boldsymbol{x})$
quantize latent elements into the discrete space $S$
    $\boldsymbol{z}_1 = Q(\boldsymbol{z}_0), \boldsymbol{z}_1 \in \boldsymbol{S}$
repeat
    calculate the objective function: $F(\boldsymbol{x}, G(\boldsymbol{z}_k))$
    gradient descent: $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \alpha \cdot \nabla F(\boldsymbol{z}_k)$
    quantize latent elements into the discrete space $S$
    $\boldsymbol{z}_{k+1} = Q(\boldsymbol{z}_{k+1}), \boldsymbol{z}_{k+1} \in \boldsymbol{S}$
until convergence to optimal latent variable $\tilde{\boldsymbol{z}}$ or maximum iteration number satisfied
apply Coding to $\tilde{\boldsymbol{z}}$
```

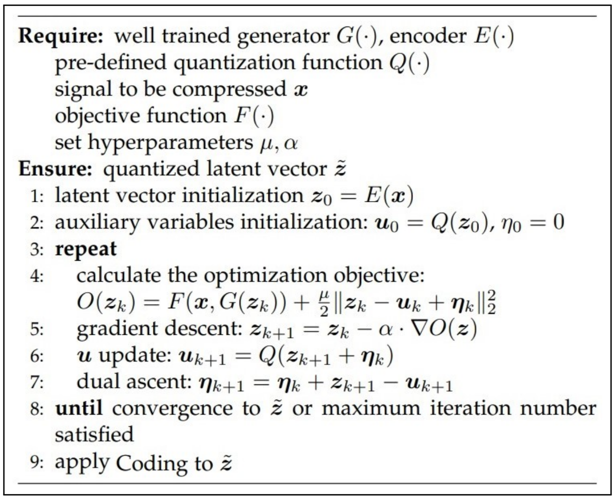
Figure 4: BPGAN Algorithm

The basic algorithm was the BPGAN Compression algorithm. Initially, the GAN-based generator network was well trained, an autoencoder was constructed; also the quantization function was pre-defined. The signal to be compressed x is given as an input along with objective function $F(.)$ and quantized set S. These prerequisites were fulfilled by the well-trained GAN initially. The BPGAN compression algorithm is shown in figure.

3.5.2 ADMM Algorithm

ADMM alternating direction method of multipliers is an algorithm that solves convex optimization problems by breaking them into smaller pieces, each of which is then easier to handle. Initially, the GAN-based generator network was well trained, an autoencoder was constructed; also, the quantization function was pre-defined. The signal to be compressed x is given to input along with objective function $F(.)$ and hyperparameters μ and α are initialized.

The objective of the algorithm is to optimize the latent vector quantization starting with latent vector initialization and quantization iteratively calculating objective function, gradient descent, and quantized latent element on discrete space until convergence to the optimal latent variable of the BPGAN. The ADMM algorithm is shown in the figure.



```
Require: well trained generator $G(\cdot)$, encoder $E(\cdot)$
    pre-defined quantization function $Q(\cdot)$
    signal to be compressed $\boldsymbol{x}$
    objective function $F(\cdot)$
    set hyperparameters $\mu, \alpha$
Ensure: quantized latent vector $\tilde{\boldsymbol{z}}$
    latent vector initialization $\boldsymbol{z}_0 = E(\boldsymbol{x})$
    auxiliary variables initialization: $\boldsymbol{u}_0 = Q(\boldsymbol{z}_0), \eta_0 = 0$
    repeat
        calculate the optimization objective:
        $O(\boldsymbol{z}_k) = F(\boldsymbol{x}, G(\boldsymbol{z}_k)) + \frac{\mu}{2}\|\boldsymbol{z}_k - \boldsymbol{u}_k + \boldsymbol{\eta}_k\|_2^2$
        gradient descent: $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \alpha \cdot \nabla O(\boldsymbol{z})$
        $\boldsymbol{u}$ update: $\boldsymbol{u}_{k+1} = Q(\boldsymbol{z}_{k+1} + \boldsymbol{\eta}_k)$
        dual ascent: $\boldsymbol{\eta}_{k+1} = \boldsymbol{\eta}_k + \boldsymbol{z}_{k+1} - \boldsymbol{u}_{k+1}$
    until convergence to $\tilde{\boldsymbol{z}}$ or maximum iteration number satisfied
    apply Coding to $\tilde{\boldsymbol{z}}$
```

Figure 5: ADMM Algorithm

3.6 Potential Verification and Validation Procedures

The quantitative performance of the proposed audio compression method is primarily measured by the quality of the speech compression. Both subjective and objective matrices can be used to evaluate the speech signal. The potential matrices could be PSEQ, human evaluation under the guidance of MUSHRA, and phoneme recognition tests.

3.6.1 PESQ and POLQA

Perceptual evaluation of speech quality PESQ is an objective metric designed to predict the mean opinion score (MOS) for speech quality by an algorithm. It is adopted by ITU-T as a recommended standard metric. This industry-standard audio quality measure considers characteristics such as audio sharpness, volume, background noise, interference, and latency in the audio signal. PESQ values range upto 4.5, with a larger score indicating better quality. The PESQ test compares the original voice with audio output creating a fully unbiased objective indicator. The PESQ score can be divided into 6 bands as follows:

Table 2: PESQ score table meaning

SN	Score	Meaning
1	1.00-1.99	No meaning
2	2.00-2.39	Considerable effort required
3	2.40-2.79	Moderate effort required
4	2.80-3.29	Small amount of effort required
5	3.30-3.79	Appreciable effort is required
6	3.80-4.50	Complete relaxation possible

POLQA Perceptual Objective Listening Quality Analysis is also an ITU-T standard and successor of the PESQ model to predict speech quality by means of analyzing digital speech signals. POLQA avoids the weaknesses of the current PESQ model and is extended toward the handling of higher bandwidth audio signals.

3.6.2 Subjective Evaluation

The ITU-R Recommendation defined subjective assessment of the intermediate audio quality level of audio signal as a method for the subjective assessment of intermediate audio quality. This method utilizes the same grading system used to assess picture quality and shares many similarities with Recommendation ITU-R BS.1116. ”MUlti Stimulus test with Hidden Reference and Anchor (MUSHA)” is the name of the technique, and it has undergone successful testing. These experiments have shown that the MUSHRA approach provides precise and dependable findings when used to evaluate intermediate audio quality. 5 people are asked to listen to the original and compressed audio clips as part of the subjective quality of the (de)compressed speech with experiment. The users are then asked to rate the test samples from 0 to 100 based on their perceptual evaluation using Multiple Stimuli with Hidden Reference and Anchor

(MUSHRA).

4. Result

This section describes the detailed procedure of model building and its results in each step. The audio compression evaluation results at intermediate stages are presented here.

4.1 Pre-processing

Here in the audio compression, the raw audio of .wav format data was first converted into mel spectrogram which is shown in the figure.

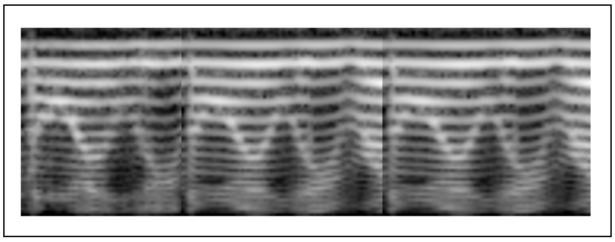


Figure 6: Mel Spectrogram

STFT was used to transform the audio signal into a spectrogram signal. Here the time domain signal is converted into the frequency domain for analysis and processing.

4.2 Encoder Initialization

The mel spectrogram is then fed to the encoder network. The encoder maps the variable length source sequence into a fixed length vector named latent vector in this way a latent vector was formed. NN autoencoder module is used to encode the mel spectrogram to the latent vector.

4.3 Optimize the Latent Vector

Here the objective is to compress the generated audio signal generated from the GAN Generator. The encoder initialized latent vector is optimized iteratively updating the latent vector according to the loss output of GAN. Here the latent vector z is updated on backpropagation through the generator of GAN. Basically, the gradient $\delta\ F(x, G(Z)) / \delta$ Z was computed for each iteration and optimal latent vector z was obtained that minimizes the loss function.

4.4 Signal Reconstruction

At the receiver side, the signal is obtained and decompressed by feeding the signal z to the same generator network from the transmitter. Before applying the signal to the generator, it was first entropy decoded and at the end, the audio signal was reconstructed from the mel spectrogram using inverse STFT on post-processing.

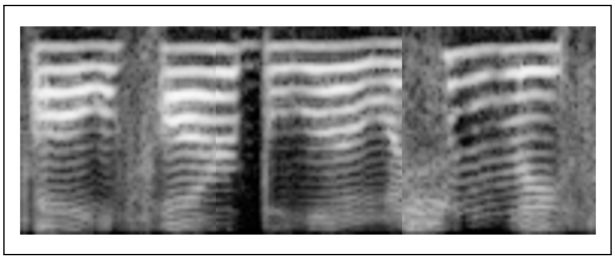


Figure 7: Generated mel spectrogram

4.5 Output

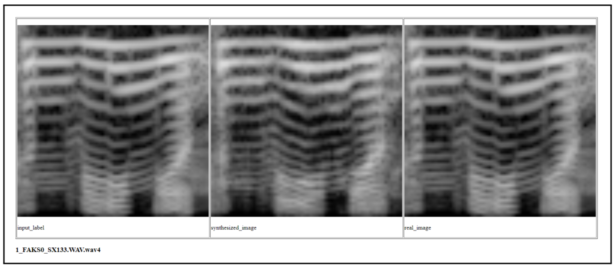


Figure 8: Original and synthesized audio spectrogram

The output of the framework is the pair of the input image and the synthesized image which is the mel spectrogram. This spectrogram was applied to the inverse STFT to construct the audio signal back. And the evaluation of the output was performed.

5. Discussion and Analysis

The model present here was built as shown in the figure for audio signal compression using GAN. First of all preprocessing of the data was performed and the data was converted into mel spectrogram form as described in section 4 of the paper which involves the STFT.

In the next step, the GAN model was initialized with the latent vector z which was constructed using an autoencoder, and the GAN model was trained for the dataset described in section 3. The model was built and trained and tested for the ADMM backpropagation algorithm. The model was trained for 160 epochs normally signify that the GAN model was stabilized therefore the training was stopped and mean square error loss of the model MSE Loss, Discriminator loss on real image D real, and discriminator loss on fake image D fake was plotted as shown in the figure below.

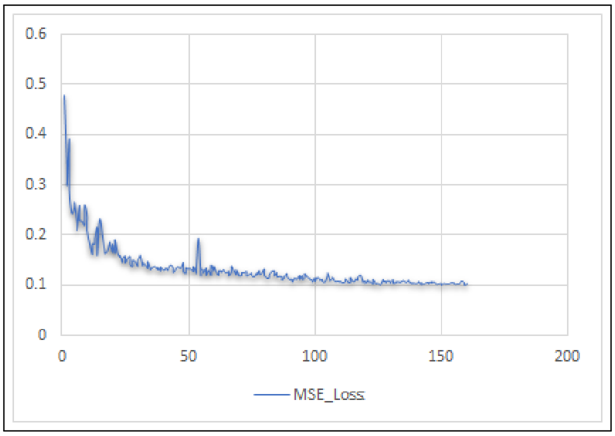


Figure 9: Mean square error loss

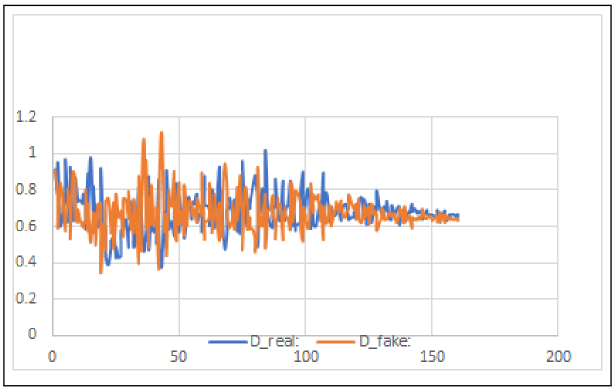


Figure 10: Discriminator loss on a real and fake image

The compressed speech rate was measured and evaluated using a latent vector z size of 512 and 16 non-uniform quantization levels for each element. Speech compression performance was present on the table below which shows the performance of the model is same as that of the Unified GAN Model performance.

Table 3 presents the compressed speech rate shows that the bitrate of the signal can be brought down to 6.6 kbps with the cost of quality matrix PESQ and MUSHRA reduction. The table suggests that balancing the compression constraints like quality and compression ratio is unable to be attained hence [26] suggests using the deep learning algorithm to compress the signal to balance the compression constraints. Hence the use of Deep learning algorithm shows that the bit rate of the signal is possible to bring down to 2 kbps with the cost of compromising quality matrix PESQ and MUSHRA reduction.

Table 3: Speech Compression Performance Comparison

Method	Bitrate	PESQ	MUSHRA
Original	256k	4.50	95.0
Unified GAN	2k	3.25	64.1
CELP	8k	3.39	59.4
Opus	9k	3.47	79.3
AMR	6.6k	3.36	58.9
BPGAN	2k	3.26	64.9

6. Conclusion

A GAN-based compression model was described, as well as a Back propagation GAN based audio signal compression approach that creates compressed data with acceptable quality at comparatively low bitrate via iterative back-propagation to find the ideal latent vector. The method uses ADMM with non-uniform quantization to look for the best latent representation of the signal in order to increase compression ratio. The method first trains the generator network model in a GAN configuration, after which it repeatedly updates and discretizes the best latent code for each signal input for compression using the pre-trained generator. Results from experiments show that compared to other techniques measured using different criteria, such as neural network-based image classification and audio phoneme recognition, the signal compressed using GAN shows a much lower data rate and acceptable signal quality. In the future, further extension of the presented approach could be enhanced by adopting the different optimization techniques instead of ADMM.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[2] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[3] Cong Leng, Zesheng Dou, Hao Li, Shenghuo Zhu, and Rong Jin. Extremely low bit neural network: Squeeze the last bit out with admm. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[4] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

[5] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *International Conference on Machine Learning*, pages 2922–2930. PMLR, 2017.

[6] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

[7] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4394–4402, 2018.

[8] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 221–231, 2019.

[9] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018.

[10] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2018.

[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7939–7948, 2020.

[12] Manfred Schroeder and B Atal. Code-excited linear prediction (celp): High-quality speech at very low bit rates. In *ICASSP'85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 937–940. IEEE, 1985.

[13] J Valin, Koen Vos, T Terriberry, and A Moizard. Rfc 6716: Definition of the opus audio codec. *Internet engineering task force (IETF) standard*, 2012.

[14] Bruno Bessette, Redwan Salami, Roch Lefebvre, Milan Jelinek, Jani Rotola-Pukkila, Janne Vainio, Hannu Mikkola, and Kari Jarvinen. The adaptive multirate wideband speech codec (amr-wb). *IEEE transactions on speech and audio processing*, 10(8):620–636, 2002.

[15] Srihari Kankanahalli. End-to-end optimized speech coding with deep neural networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2521–2525. IEEE, 2018.

[16] Milos Cernak, Alexandros Lazaridis, Afsaneh Asaei, and Philip N Garner. Composition of deep and spiking neural networks for very low bit rate speech coding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12):2301–2312, 2016.

[17] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[18] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. In *International Conference on Machine Learning*, pages 2410–2419. PMLR, 2018.

[19] W Bastiaan Kleijn, Felicia SC Lim, Alejandro Luebs, Jan Skoglund, Florian Stimberg, Quan Wang, and Thomas C Walters. Wavenet based low rate speech coding. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 676–680. IEEE, 2018.

[20] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[21] Cristina Gârbacea, Aäron van den Oord, Yazhe Li, Felicia SC Lim, Alejandro Luebs, Oriol Vinyals, and Thomas C Walters. Low bit-rate speech coding with vq-vae and a wavenet decoder. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 735–739. IEEE, 2019.

[22] Santiago Pascual, Antonio Bonafonte, and Joan Serra. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

[23] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

[24] Andrés Marafioti, Nathanaël Perraudin, Nicki Holighaus, and Piotr Majdak. Adversarial generation of time-frequency features with application in audio synthesis. In *International conference on machine learning*, pages 4352–4362. PMLR, 2019.

[25] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

[26] Bowen Liu, Ang Cao, and Hun-Seok Kim. Unified signal compression using generative adversarial networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3177–3181. IEEE, 2020.