GAN-Based Bandwidth Extension for Music

Cassius Close
University of Rochester
cclose@ur.rochester.edu

ABSTRACT

Bandwidth extension is the process of increasing the bandwidth of audio signals: generally, it creates high frequency content not previously existent in a signal. It is an important task in the problems of music restoration and enhancement and speech enhancement for low data-rate communication media. Among the recent neural methods applied to bandwidth extension, general adversarial networks (GANs) provide high quality results and a breadth of existing techniques and literature. However, the vast majority of existing methods are designed to work on speech, and the research towards bandwidth extension of music is somewhat limited. The methods that are designed for music focus on a very specific genre, and do not generalize well. The proposed method uses techniques from recent high-quality GAN models designed for speech to try to create a model for music bandwidth extension that generalizes well to different genres. The results are a good start towards general music bandwidth extension, but much work could be done to improve these methods to compete with state of the art models for speech.

1. INTRODUCTION

Audio bandwidth extension (BWE) is the process of increasing the bandwidth of an audio signal. Most of the time, this is applied to audio recorded at a low sample rate. The sample rate is an upper limit on the frequencies that can be represented in an audio recording, so the task is usually to increase the sample rate and generate the high frequency content previously unavailable.

BWE is usually split up into two domains: speech and music. The most applicable task of BWE of music is to improve the quality of old recordings, where the sample rate was limited by the recording technology of the time. For speech, the most popular task is improving the quality of telephone signals, which still work at low sample rates for low-resource data transfer. Older telephone networks used a sample rate of 3.4kHz, and modern systems use 8kHz (which is still quite low) [6] [7], making the problem of bandwidth extension highly relevant for making speech intelligible over the phone.

Because the task of speech enhancement is a more practical problem, much more research has gone into bandwidth extension of speech than of music. The domain of speech is simpler than that of music, which can vary widely depending on genre and instrumentation.

I am interested in the task of enhancing old musical recordings, so I chose to focus on music instead of speech.

2. RELATED WORK

Bandwidth extension has been studied for a long time (for example, this source-filter model-based method from 1979 [8]). In the last 15 years, researchers have begun studying neural networks for use with this task. Early methods used feedforward networks or convolutional neural networks (CNNs) [6] [7] [9]. Recently, generative models have achieved much improvement in the quality of bandwidth-extended signals. These models include variational auto-encoders, generative adversarial networks, flow-based models, and diffusion models. Flow and diffusion models are the newest approaches, and they seem to produce very high-quality results [10] [11]. Because they are so new, there is not as much existing work using them, especially in the field of bandwidth extension.

Generative Adversarial Networks (GANs) are another type of generative model that produces quality results. They have a wide literature of methods and techniques built up around them. There are two GAN architectures that seem to work very well for speech synthesis tasks: WaveNet (WaveNet [12] and later, HiFi-GAN [13]), and U-Net [14]. BWE is very similar to speech synthesis, so it is not surprising that several papers have adapted these methods to BWE, including HiFi-GAN+ [2] and BEHM-GAN [1].

Of the GAN-based methods, the vast majority work on speech [2] [6] [7] [8] [9]. I have found three recent papers that focus on music [16] (including MU-GAN [15] and BEHM-GAN [1]). Two of these restrict their datasets to solo classical piano recordings, which is a simpler domain to work in. When applied to different genres or instruments, the accuracy of these methods decreases [1] [15]. The third method trains on a variety of popular genres, but the results have noticeable artifacts [16].

Notably, BEHM-GAN is the only paper that has actually performed BWE on historical recordings. Though the domain is limited to solo piano, the results are very high quality [1].

Interestingly, these three methods all use the U-Net architecture. Based on the success of HiFi-GAN [13] for speech synthesis and HiFi-GAN+ [2] for BWE, this proposed method follows the WaveNet architecture with the goal of performing high quality BWE that generalizes well to a variety of genres.

3. METHODS

The proposed method takes a mono audio signal sampled at 16kHz (with a bandwidth of 8kHz) and increases the sample rate to 44.1kHz (with a bandwidth of 22.05kHz). It

uses a WaveNet-based time-domain GAN to generate the new frequency content between 8kHz and 22.05kHz.

Generative Adversarial Networks (GANs) use two types of neural networks that compete with each other. A generator network creates the data; in this case, this is the full-bandwidth audio. Often, this is a convolution neural network (CNN) that maps the low-bandwidth input to a full-bandwidth output. One or multiple discriminator networks attempt to determine whether an input is real or created by the generator. These are also usually CNNs, though with different architecture than that of the generator. The two types of networks are trained at the same time: the discriminator tries to guess if its inputs are real or fake, and the generator tries to generate data that can fool the discriminator. As one improves its performance, so will the other.

GANs work well for audio synthesis because they generate highly detailed output. Standard deep-learning models (CNN, etc.) tend to create audio without much detail – resulting in an "oversmoothed" spectrum [2] [1]. The presence of the discriminators encourages the generator output to be more detailed, like the ground truth, to better fool the discriminators.

This method chooses to work in the time-domain instead of the time-frequency domain. This means that instead of the model inputting and outputting a spectrogram, it works with raw waveforms. Many BWE methods output a magnitude spectrogram, which means they must find a way to generate phase to convert the spectrogram back to an audio signal. Working directly in the time domain is convenient, because the phase information is stored implicitly in the audio samples.

3.1 Generator Architecture

The generator's job is to take low-bandwidth audio as input and to output full-bandwidth audio. Usually, generators follow the convolution neural network (CNN) architecture. In a CNN, each layer is connected to a fixed number (called the kernel size) of consecutive nodes in the previous layer. In figure 1, the kernel size is 3, so each node in the first convolution layer is connected to 3 consecutive input samples. Each node in the second layer is connected to 3 nodes in the first layer, each of which is connected to 3 input samples. Thus, in the case of audio, the deeper layers in the network can model longer temporal relationships between samples that aren't as close to one another (a larger "receptive field").

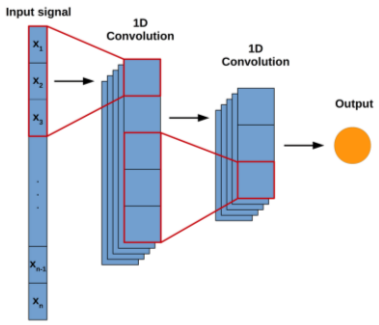


Figure 1. A visualization of convolution from researchgate.net [17]. Each node in a layer is connected to a given number of consecutive nodes in the previous layer.

This method is based on the WaveNet generator architecture, which consists of layers of dilated convolutions applied to the waveform. Instead of connecting to consecutive nodes, dilated convolution skips nodes in between its connections. In figure 2, the second layer has a kernel size of 2 and a dilation of 2, which means that it connects to every other node in layer 1 (for the length of 2 nodes). This allows longer temporal relationships to be modeled with the same number layers as with normal convolution.

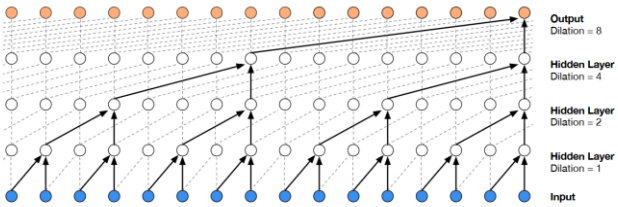


Figure 2. A visualization of the large "receptive field" of dilated convolutions, from the Google WaveNet paper [12]. Normal convolutional layers overlap in their windows, but with dilation, the nodes in each next layer are connected to an exponentially increasing number of input nodes.

The WaveNet diagram in figure 2 is causal, which means that an output sample only depends on current and past input samples. The proposed method uses non-causal convolution, which means that the output sample depends on an equal number of past and future input samples. In the diagram, this would appear as the architecture mirrored to the right of the current sample.

The proposed method uses 5 layers of dilated convolution with a kernel size of 3 and dilation increasing by a factor of 3. A greater number of layers would allow for a greater receptive field but was limited by memory constraints.

3.2 Discriminators

The discriminator's job is to try to tell the difference between ground truth and generated outputs. It is usually some form of CNN that outputs a single number: how sure it is that the input is real. This could be a probability, but it doesn't have to be. A higher number means the discriminator is more sure that the input is ground truth. As the discriminator gets better at classifying its inputs, its results are used to train the generator, so that the generator can create outputs that better fool the discriminator. This is called adversarial training, because the two networks are competing against each other.

Following cues from HiFi-GAN [13], the discriminators use multiple strided convolution layers. As with standard convolution, each node in a strided convolution layer connects to a kernel of consecutive nodes in the previous layer. In standard convolution, each next node in the convolution layer shifts the kernel by one node in the previous layer. In strided convolution, each next node in the convolution layer shifts the kernel by multiple nodes in the previous layer. Figure 3 shows this. The kernel in the first layer is shifted by a stride of 2, which results in a 2x2 convolution layer. In normal convolution, the next layer would

have been 3x3. Strided convolution layers are used to decrease the size of layers quickly, which is desired because the discriminators need to output one number.

The last strided convolution layer in each discriminator is averaged to output a single number.

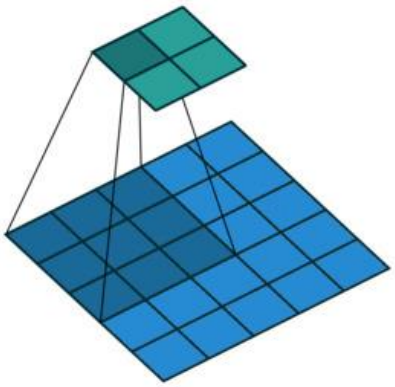


Figure 3. A visualization of strided convolution from [18]. The kernel for each node in a convolution layer is shifted by more than one node in the previous layer.

Recent methods have found that using several discriminators of different scales and domains improve the quality of the generated audio and removes different kinds of artifacts [1] [2] [13]. These methods use discriminators in both the time domain (on the waveform) and the time-frequency domain (on the spectrogram). For each domain, there can be multiple discriminators at different scales. For the waveform, this means discriminators on the output audio, as well as progressively down-sampled versions. For the spectrogram, this means discriminators on spectrograms with different window and hop lengths. These different scales encourage the generator to create details at different temporal resolutions.

The proposed method uses a single spectrogram discriminator with a window size of 2048 and a hop size of 512. It uses three waveform discriminators that work on the raw waveform, and 2x and 4x downsampled version of the waveform. Downsampling is performed with average pooling. More on these discriminators in the results section.

3.3 Loss

To train neural networks, we try to minimize some loss function that attempts to measure the error in the network's output. The smaller the error/loss, the better the results.

GANs combine several loss functions for better training. First, the generators use standard loss functions you might find in normal CNNs, called feature loss. The proposed method uses the L1 distance between the generated and ground truth waveform and spectrogram. Minimizing this means the generator's output will be closer to the ground truth.

The discriminators should output a large number when their input is real, and a small number when the input is fake. Thus, to train the discriminators, this method uses the loss function in equation 1, where D(x) is the output of the discriminator given input x. Minimizing the loss will encourage small numbers for fake input and large numbers for real input.

$$DLoss = relu\big(1 + D(fake)\big) + relu(1 - D(real)) \quad (1)$$

To use the information from the discriminators to affect the generation of data, we must include the discriminator output somewhere in the generator's loss function. This is called adversarial loss and is shown in equation 2. When training the discriminators, we want to improve the quality of the discriminator output, so we minimize 1 + D(fake). When training the generator, we want to hurt the performance of the discriminator, so we include -D(fake).

$$AdvLoss = -D(fake) \quad (2)$$

The generator's total loss function is the sum of the feature loss and adversarial loss function.

3.4 Training

This method uses the DSD100 dataset, which contains 100 recordings of music in popular genres [19]. 80 recordings are used for training, and 20 recordings are used for validation/testing.

It is a common technique with GANs to first train the generator on its own with a higher learning rate, and then to train with the discriminator losses at a lower learning rate [1] [2]. This lets the training quickly get to roughly the right place (a smoothed spectrum) and then use the discriminators to encourage the generation of fine details.

It is also common to run the discriminator training twice for each generator output [1] [2].

The proposed method trains using a batch size of 3 due to memory constraints. It uses the Adam optimizer and trains the generator with just feature loss at a learning rate of 1e-3 for 189 iterations. It then includes adversarial loss and trains at a learning rate of 1e-5 for 255 iterations.

4. RESULTS

To determine the effect of different parts of the model, testing was performed with three different versions of the model: generator only (no discriminators), generator + waveform discriminators, and generator + waveform and spectrogram discriminators.

4.1 Measuring Performance

4.1.1 Perceptual Measures

The most accurate way to measure the perceptual quality of audio synthesis (i.e. does the audio sound "good quality", artifact-free, etc. to humans) is to perform listening tests. In these tests, participants are asked to blindly rank the quality of several audio samples. Among these samples are the ground truth recordings, the results of the method being tested, and results of several other methods to evaluate performance of the proposed method. Any sort of listening test is beyond the scope of this project, and thus I must rely on objective measures to evaluate the performance.

4.1.2 Objective Measures

There are several classic objective measures that are simple to calculate and very common in evaluating audio quality. These are Signal-to-Noise Ratio (SNR), Signal-to-Distortion Ratio (SDR), and Log-Spectral Distortion (LSD). Even though these measures are quite common, it has been shown that they are not very representative of perceptual quality [1] [2]. The intuition offered is that GANs learn to create realistic details in the high frequencies, whether or not they exactly match the ground truth content. An over-smoothed spectrum might be closer to the ground truth and thus have better results in the above measures but will sound worse to human ears [2].

To improve the perceptual quality of objective measures, a number of neural-based measures have been created that attempt to approximate human judgement [3] [4] [5]. Unfortunately, none of them are very appropriate for this task. Many of them are designed to evaluate speech [5] or have other qualities that make them unsuitable, so we are left to rely on the unreliable objective measures.

Most BWE methods include some combination of SNR, SDR, and LSD in their measures, so I measured all three of them.

4.1.3 Measurements

Table 1 shows the objective results measured for the three different versions of the method. These results do not seem to consistently improve or worsen. For example, SNR decreases when the waveform discriminators are added, but then improves again when the spectrogram discriminator is added. As a counterexample, SDR increases with the waveform discriminators, but decreases with the spectrogram discriminator. My takeaway is that these results are not very conclusive.

	SNR	SDR	LSD
Generator Only	8.44 dB	13.91 dB	3.65 dB
Waveform Discriminators	7.45 dB	15.35 dB	2.64 dB
All Discriminators	8.45 dB	14.87 dB	2.59 dB

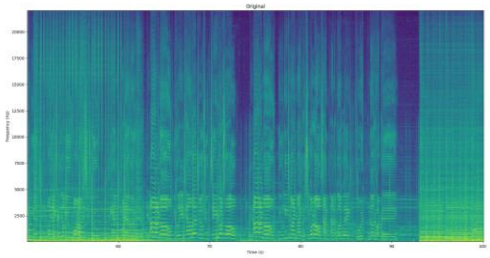
Table 1. Results of objective measures for the proposed method.

4.2 Informal Results

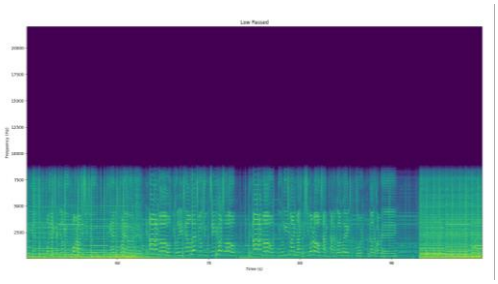
Figure 4 displays the spectrograms of a test example passed through each version of the model. With large enough figures, one can see the harmonic combs from the low frequency spectrum have been extended into the higher frequency range. Each progressive addition of discriminators adds more high frequency content that brings it closer to the ground truth.

In listening to the output, the change in high frequency content is noticeable. The recordings sound more noticeably more wideband and more "present" than the downsampled input. However, when compared to the ground truth, the reconstructed audio still sounds much more narrow-band.

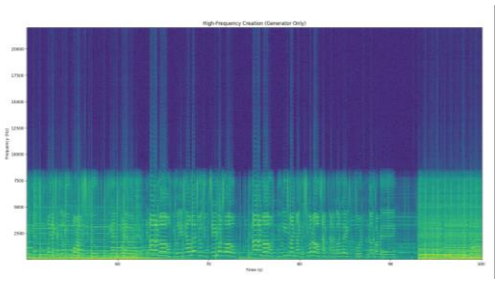
Interestingly, I found that even though adding the spectrogram discriminator caused spectrogram to look more similar to the ground truth, the high frequency content was noticeably distorted in a way that it was not when only the waveform discriminators were used. This is supported by the SDR measurements decreasing with the addition of the spectrogram discriminator. I am not sure why this is.



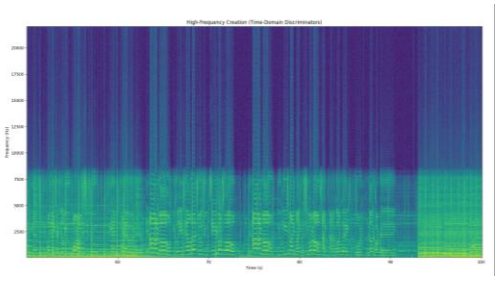
a) Ground truth wide-band signal



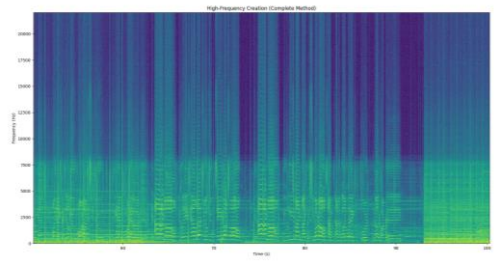
b) Low-passed input signal



c) Results for the generator only



d) Results for the generator + waveform discriminators



e) Results for the generator + all discriminators

Figure 4. Sample test results of the different versions of the models. One can see that as discriminators are added, the output becomes closer to the ground truth.

5. CONCLUSIONS

These results are a decent beginning towards genre-generalized music bandwidth extension. Obviously, a lot of work needs to be done to improve the results, as they do not measure up to the state-of-the-art methods. There are two explanations I have for this.

First, the domain of "music" is a difficult one to work with. It is a lot broader, a lot more varied, than the domains of speech or solo piano, which most existing bandwidth extension methods work on. Because it is a wider domain, training a model to generalize well needs a lot of data, and the dataset used had only 100 recordings. With significantly more data, I believe generalizability would improve.

Secondly, training was severely limited by memory constraints. With more memory, training could be conducted with larger batch sizes, and the models could have more layers, larger receptive fields, etc., all of which would hopefully improve results.

This being said, I am happy with my results, as this is my first real foray into the world of machine learning.

6. REFERENCES

[1] Moliner, Eloi, and Vesa Välimäki. "BEHM-GAN: Bandwidth Extension of Historical Music Using Generative Adversarial Networks." arXiv, June 28, 2022. http://arxiv.org/abs/2204.06478.

[2] Su, Jiaqi, Yunyun Wang, Adam Finkelstein, and Zeyu Jin. "Bandwidth Extension Is All You Need." In ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 696–700. Toronto, ON, Canada: IEEE, 2021. https://doi.org/10.1109/ICASSP39728.2021.9413575.

[3] Kilgour, Kevin, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. "Fréchet Audio Distance: A Metric for Evaluating Music Enhancement Algorithms." arXiv, January 17, 2019. http://arxiv.org/abs/1812.08466.

[4] Gemmeke, Jort F., Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. "Audio Set: An Ontology and Human-Labeled Dataset for Audio Events." In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 776–80. New Orleans, LA: IEEE, 2017. https://doi.org/10.1109/ICASSP.2017.7952261

[5] Rix, A.W., J.G. Beerends, M.P. Hollier, and A.P. Hekstra. "Perceptual Evaluation of Speech Quality (PESQ)-a New Method for Speech Quality Assessment of Telephone Networks and Codecs." In 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221), 2:749–52. Salt Lake City, UT, USA: IEEE, 2001. https://doi.org/10.1109/ICASSP.2001.941023.

[6] Kuleshov, Volodymyr, S. Zayd Enam, and Stefano Ermon. "Audio Super Resolution Using Neural Networks." arXiv, August 2, 2017. http://arxiv.org/abs/1708.00853.

[7] Kontio, Juho, Laura Laaksonen, and Paavo Alku. "Neural Network-Based Artificial Bandwidth Expansion of Speech." IEEE Transactions on Audio, Speech and Language Processing 15, no. 3 (March 2007): 873–81. https://doi.org/10.1109/TASL.2006.885934.

[8] Makhoul, J., and M. Berouti. "High-Frequency Regeneration in Speech Coding Systems." In ICASSP '79. IEEE International Conference on Acoustics, Speech, and Signal Processing, 4:428–31. Washington, DC, USA: Institute of Electrical and Electronics Engineers, 1979. https://doi.org/10.1109/ICASSP.1979.1170672.

[9] Li, Kehuang, Zhen Huang, Yong Xu, and Chin-Hui Lee. "DNN-Based Speech Bandwidth Expansion and Its Application to Adding High-Frequency Missing Features for Automatic Speech Recognition of Narrowband Speech." In Interspeech 2015, 2578–82. ISCA, 2015. https://doi.org/10.21437/Interspeech.2015-555.

[10] Kong, Zhifeng, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. "DIFFWAVE: A VERSATILE DIFFUSION MODEL FOR AUDIO SYNTHESIS," 2021, 17.

[11] Han, Seungu, and Junhyeok Lee. "NU-Wave 2: A General Neural Audio Upsampling Model for Various Sampling Rates." In Interspeech 2022, 4401–5, 2022. https://doi.org/10.21437/Interspeech.2022-45.

[12] Oord, Aaron van den, 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, September 19, 2016. http://arxiv.org/abs/1609.03499.

[13] Kong, Jungil, Jaehyeon Kim, and Jaekyoung Bae. "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis." arXiv, October 23, 2020. http://arxiv.org/abs/2010.05646.

[14] Stoller, Daniel, Sebastian Ewert, and Simon Dixon. "Wave-U-Net: A Multi-Scale Neural Network for

End-to-End Audio Source Separation." arXiv, June 8, 2018. http://arxiv.org/abs/1806.03185.

[15] Kim, Sung, and Visvesh Sathe. "Bandwidth Extension on Raw Audio via Generative Adversarial Networks." arXiv, March 21, 2019. http://arxiv.org/abs/1903.09027.

[16] Hu, Shichao, Bin Zhang, Beici Liang, Ethan Zhao, and Simon Lui. "Phase-Aware Music Super-Resolution Using Generative Adversarial Networks." arXiv, October 9, 2020. http://arxiv.org/abs/2010.04506.

[17] A Novel Deep Learning Model for the Detection and Identification of Rolling Element-Bearing Faults - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Simple-1D-convolutional-neural-network-CNN-architecture-with-two-convolutional-layers_fig1_344229502

[18] Dumoulin, Vincent, and Francesco Visin. "A Guide to Convolution Arithmetic for Deep Learning." arXiv, Janu-ary 11, 2018. http://arxiv.org/abs/1603.07285.

[19] A. Liutkus et al., 'The 2016 Signal Separation Evaluation Campaign', in Latent Variable Analysis and Signal Separation - 12th International Conference, LVA/ICA 2015, Liberec, Czech Republic, August 25-28, 2015, Proceedings, 2017, pp. 323–332.