Snare Drum Generation & Enhancement

Dalton Davis, Abhijeet Rastogi, Jun Wu

1. Introduction

1.1 Problem Statement

In this project, we will use machine learning to generate realistic, high-quality snare drum samples. We use a GAN-based and diffusion-based model (waveGAN) (diffWave) to generate the drum samples. Due to hardware constraints (GPU RAM), the generated samples were downsampled from 44.1 kHz to 22.05 kHz. A custom audio signal processing architecture was developed to reconstruct the original resolution for the generated samples. Our goal is to generate drum sounds that are comparable to existing commercial samples, enabling digital music producers to access a wider variety of high-quality drum samples without the financial burden of purchasing large and costly sample packs.

1.2 Motivation

In the digital music production industry, purchasing drum samples is common practice. However, producers often encounter issues of high costs and unpredictability. Professional-quality drum sample packs can be expensive, adding financial strain on producers and musicians who often have limited resources. A machine learning-based solution could provide a customizable and efficient alternative, allowing producers to generate drum samples on demand. This project addresses the potential to give access to diverse drum samples for all, reducing costs and allowing for creative control over sound design in music production.

2. Related Work

2.1 WaveGAN

One of the machine learning models that we used to generate drum samples was WaveGAN, a Generative Adversarial Network (GAN) based model. GAN is known to have two neural networks, a generator and a discriminator. The two networks are in a competitive relationship where the generator will be trained to produce more realistic synthetic data based on the training data while the discriminator will be trained to identify fake data generated by the generator. WaveGAN adopted the idea of GAN and it was designed for raw audio synthesis. It can generate raw audio waveforms directly, without using audio representations such as spectrograms. WaveGan has been successful in generating short audio clips, which align well with the nature of drum samples.

2.2 DiffWave

Another machine learning model we used was DiffWave, a diffusion-based generative model that has shown promising results in audio generation recently. The idea of the diffusion model is to simulate a process where noise is gradually added to training data, then the model learns to remove the noise to recover the original data. After the model was trained, it could generate data starting from random noise and remove the noise, and then what is left will be

data generated. DiffWave uses the idea of the diffusion model for audio generation. It iteratively refines noise to generate audio and has the potential to capture complex audio textures.

3. Dataset

A dataset of 5,844 snare drum audio samples of 44.1 kHz sample rate is used for training the WaveGan and DiffWave, and it acts as a “target label” for the audio upsampling signal processing architecture. The dataset consists of 57 different snare drums hit in various ways.

4. Methodology

4.1 Overview of Approach

First, we fine-tune and train available WaveGan and DiffWave to generate drum samples using the dataset. Both WaveGAN and DiffWave were trained for 800 epochs. However, because of the hardware limitations, the generated drum samples only have 22.05 kHz (22050 sample rate) instead of 44.1 kHz. Because of the Nyquist theorem, digital audio can only express frequencies up to half the sample rate, meaning that all frequencies above 11.025 kHz are lost with a sample rate of 22.05 kHz. To further improve the quality of the generated samples, we implement a custom audio signal process architecture to upsample the generated samples. We used the dataset as a “target label” to train the architecture. The architecture attempts to learn how to transform 22.05 kHz drum samples to the original 44.1 kHz, filling in the missing harmonic content lost from downsampling. After training, we passed the generated samples into the model. This reconstructed high-frequency information allows for the perception of a better quality of sound. In the end, we compute the LSD score and the MCD score of the original samples compared with the same files upsampled from the 22.05 kHz versions to achieve similar quality compared to the original samples. Additionally, we compute the probability distribution of the generated samples, upsampled samples, and training data, and then compare the difference between them to evaluate the quality of DiffWave, WaveGAN, and our Upsampling model.

4.2 Upsampling Model

The frequency-aware audio upsampling model converts low-resolution audio signals (22.05 kHz) into high-resolution versions (44.1 kHz), targeting drum sample data where preserving spectral fidelity and perceptual quality is critical. It combines time-domain and frequency-domain processing to address challenges across different frequency bands, using paired low- and high-resolution datasets for training.

The architecture includes preprocessing through a DrumSampleDataset class, which normalizes, pads, and trims audio to ensure consistent input and output lengths while generating low-resolution counterparts. The core model, the Frequency-Aware Upsampling Network, features a sub-bass suppression branch for reducing low-frequency artifacts, a time-domain branch for temporal patterns, and a spectral processing branch using Short-Time Fourier Transform (STFT) for frequency-domain features. A FrequencyBandSplit module applies targeted processing to sub, low, mid, and high-frequency bands. Outputs from these branches are fused using residual blocks and transposed convolutions, with a high-frequency

enhancement branch refining the final result.

The training uses a combination of L1 loss for time-domain alignment and a Frequency-Aware Spectral Loss that prioritizes perceptually significant mid and high frequencies. Optimization is handled with the Adam optimizer, gradient clipping, and a learning rate scheduler, with early stopping to prevent overfitting. The model was trained for 100 epochs.

5. Evaluation Metrics

5.1 Integral Probability Metrics (IPMs)

To evaluate the similarity between the distributions of the high-quality training data and the generated audio samples, we first constructed probability distributions based on the frequency spectra of the data. For the generated audio, we used a batch of 100 samples for the distribution. This was achieved by performing a 4096-point Fast Fourier Transform (FFT) on each sample to extract its frequency components. The resulting spectra were then aggregated to build the distributions. Using these distributions, we calculated two integral probability metrics: Kullback-Leibler (KL) Divergence and Wasserstein Distance.

Kullback-Leibler (KL) Divergence is a statistical measure that quantifies how one probability distribution P differs from a second reference probability distribution Q. It measures the "distance" between distributions in terms of their information content. A lower score indicates that the two distributions are similar.

$$KL(P||Q) = \sum P(x) log\left(\frac{P(x)}{Q(x)}\right)$$

Wasserstein Distance measures the cost of transforming one distribution into another. It considers both the differences in value and the spatial arrangement of distributions. It is particularly useful in comparing continuous distributions and is widely used in generative models, such as GANs, to evaluate how well a model generates samples similar to the reference data.

$$W(P,Q) = \int_{-\infty}^{\infty} \left|F_P(x) - F_Q(x)\right| dx$$ $F_P(x)$ and $F_Q(x)$ are the CDFs of $P(x)$ and $Q(x)$

5.2 Log-Spectral Distance (LSD)

Log-Spectral Distance (LSD) was used to evaluate how well the upsampling model preserved the frequency-domain characteristics of the audio signals. It measures the difference in power between the frequency spectra of two signals on a logarithmic scale, quantifying their similarity in terms of spectral energy distribution. This metric provides a direct assessment of the spectral fidelity of the generated signals, ensuring that the key spectral features of the high-quality reference data are retained. A lower LSD value indicates that the frequency components of the upsampled audio closely match those of the original signal.

5.3 Mel Cepstral Distance (MCD)

Mel Cepstral Distance (MCD) complements Log-Spectral Distance (LSD) by focusing on the perceptual quality of audio signals. It evaluates how well the upsampled audio aligns with human auditory perception by comparing mel-frequency cepstral coefficients (MFCCs), which

are derived from the human auditory scale to mimic the ear's sensitivity to different frequency ranges. This makes MCD a useful metric for assessing the perceived audio quality of generated signals. Lower MCD values indicate that the upsampled audio sounds more natural and closer to the high-quality reference data, reflecting minimal perceptual differences.

6. Results

The results found in Table 1 indicate that the effectiveness of the upsampling process varies between DiffWave and WaveGAN and depends on the evaluation metric used. For DiffWave, upsampling led to a slight improvement in KL divergence (-0.0069), suggesting a closer match between the generated and high-quality samples in terms of distribution overlap. However, the Wasserstein distance increased significantly (+620.3296), indicating that the upsampled outputs diverged from the high-quality samples in terms of overall distribution geometry. In contrast, WaveGAN showed consistent improvements across both metrics, with a noticeable reduction in KL divergence (-0.0234) and a significant decrease in Wasserstein distance (-194.7256). This reflects an enhanced alignment between the upsampled outputs and the high-quality samples. We suspect that the poor performance for the Wasserstein distance is caused by the large amount of bass frequencies found in the DiffWave generated samples that skew the distribution.

Table 1: Integral Probability Metric Comparison

	Integral Probability Metric (IPM)					
	DiffWave Output vs. HQ samples	DiffWave Upsampled vs. HQ samples	Δ	WaveGAN Output vs. HQ samples	WaveGAN Upsampled vs. HQ samples	Δ
KL Divergence	0.1835	0.1766	-0.0069	0.1296	0.1062	-0.0234
Wasserstein Distance	1164.6888	1785.0184	+620.3296	600.6578	405.9322	-194.7256

Figure 1: Frequency Distribution - DiffWave output vs. High-quality samples

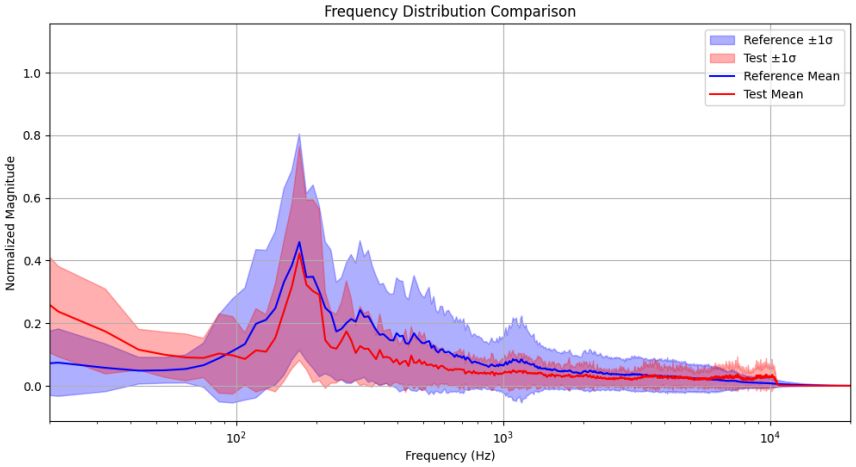


Figure 2: Frequency Distribution - Wavegan output vs. High-quality samples

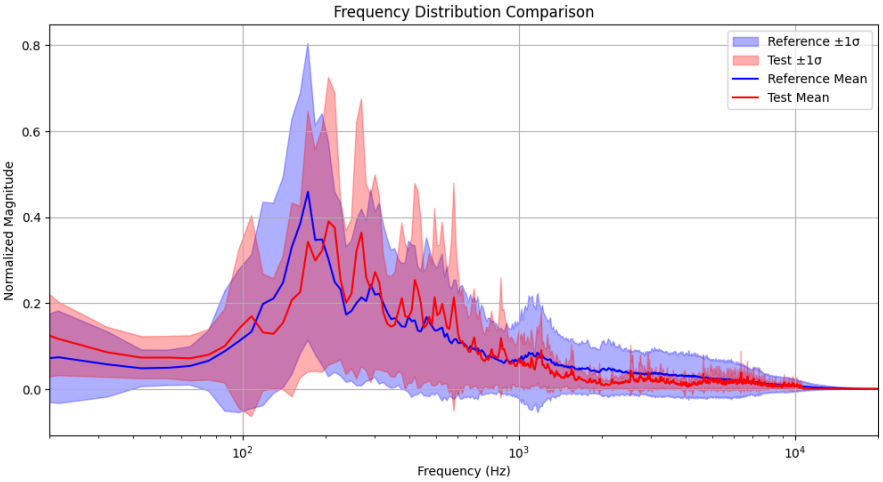


Figure 3: Frequency Distribution - DiffWave output upsampled vs high-quality samples

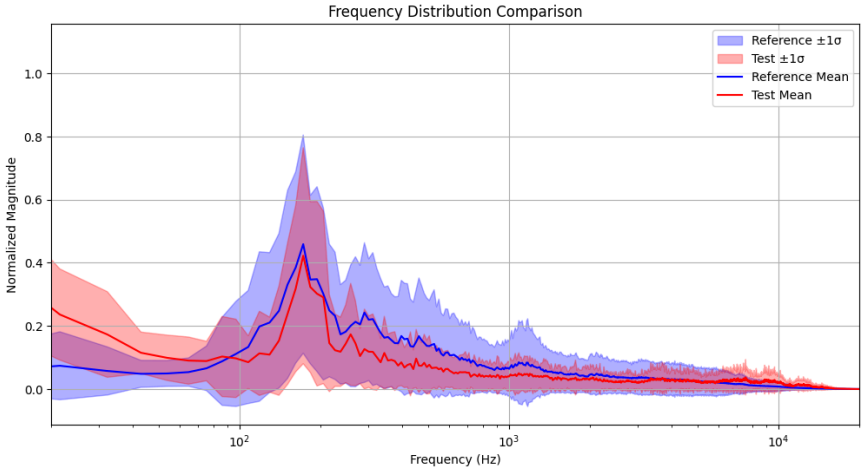
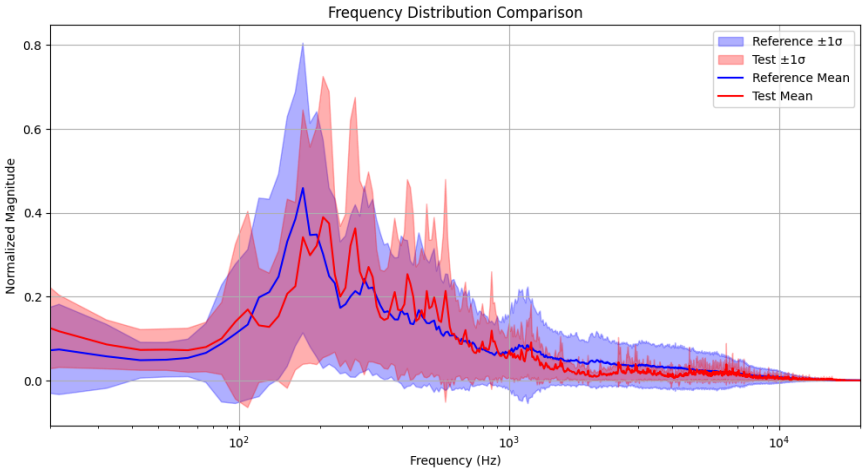


Figure 4: Frequency Distribution - Wavegan output upsampled vs high-quality samples



6.1 Quantitative Results

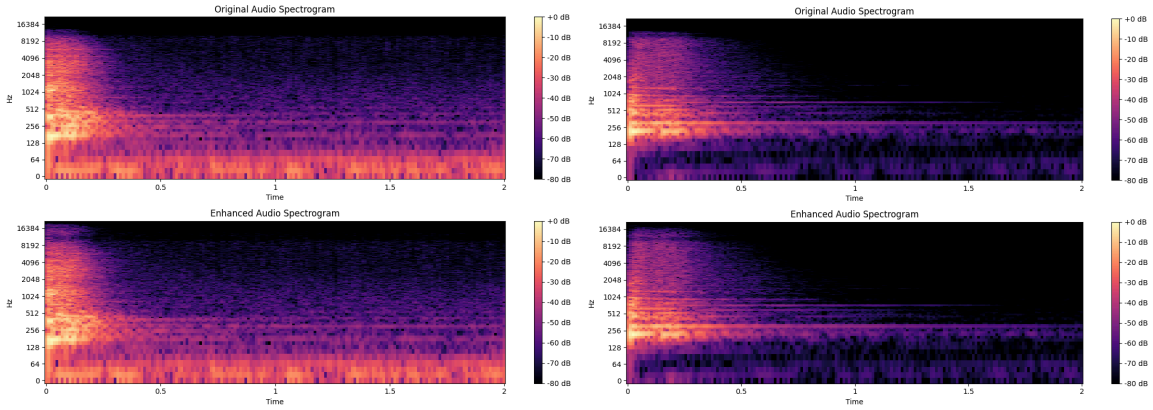
Table 2: Performance of Upsampling on five randomly selected samples

Metric	Log Spectral Distance	Mel Cepstral Distance
Sample1	0.0016	56.5998
Sample2	0.0026	93.5142
Sample3	0.0015	57.0424
Sample4	0.0013	80.8516
Sample5	0.0013	50.5280
Average	0.0017	67.7072

The Upsampling model scored below 0.01 for the Log Spectral Distance (see Table 2) which indicates that the frequency content of the upsampled audio is nearly identical to the original audio files. However, an average Mel Cepstral Distance score of 67.7072 and a range of 50.5280-93.5142 suggests that there are significant perceptual differences between the original audio and our upsampled versions. This is most likely due to the artifacts introduced by downsampling that the model was unable to reverse.

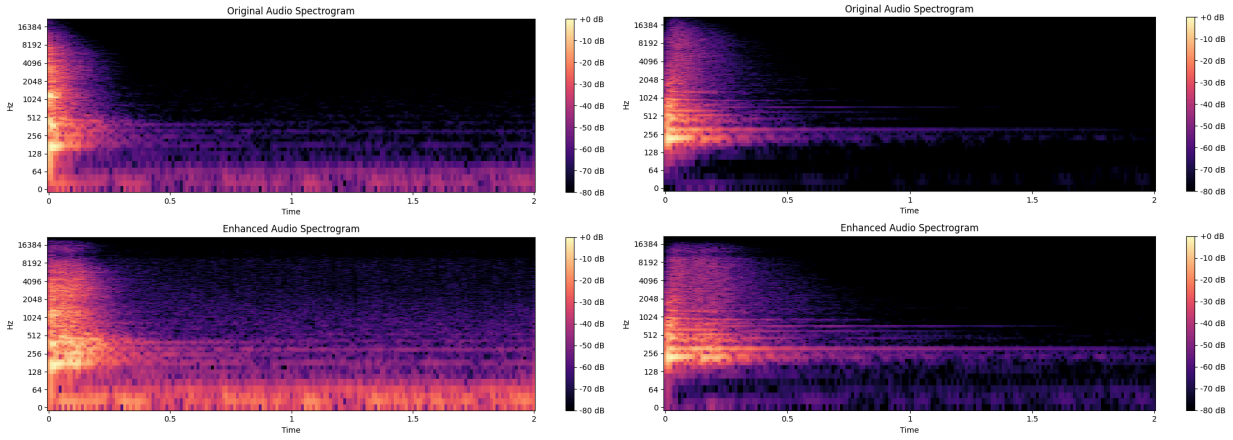
6.2 Qualitative Results

Figure 5: Before (22.05 kHz) and after(44.1 kHz) upsampling:



As seen in Figure 5 above, the high-frequency content above 11kHz is restored when upsampling from the 22.05 kHz audio. However, it was unable to get rid of the artifacts introduced from the downsampling of the original audio files (figure 6 below).

Figure 6: Original audio (44.1kHz) vs. upsampled audio (44.1kHz):



7. Conclusion

7.1 Comparison of Models

DiffWave took a total of ~30 hours to complete while WaveGAN took only ~8 hours, implying that in regards to power consumption and convenience WaveGAN performs better. Overall, WaveGAN received better scores than DiffWave for both the KL divergence and the Wasserstein distance which indicates that the samples generated from WaveGAN more closely match the frequency distribution of the training data. However, DiffWave produces a wider variety of sounds compared to WaveGAN despite being trained on the same data.

7.2 Challenges Faced

- Hardware Limitations: The generated samples were downsampled to 22.05 kHz due to hardware constraints, resulting in the loss of high-frequency content.
- Training Stability: WaveGAN initially produced silent outputs due to an overly large learning rate. Adjusting the learning rate from 10e-4 to 10e-5 resolved this issue.
- Downsampling Artifacts: While the Upsampling architecture restored high-frequency content above 11 kHz, we were unable to eliminate artifacts caused by the downsampling process which would need further tuning of the model to accomplish.
- Change in loss function: In the first version of our Upsampling model, the training and validation loss was very unstable and did not improve after ~70 epochs. We changed our loss function to the frequency band L1 loss function which improved training stability and produced better results overall. (see figures 7 and 8 below)

Figure 7: Upsampling loss plot - First version

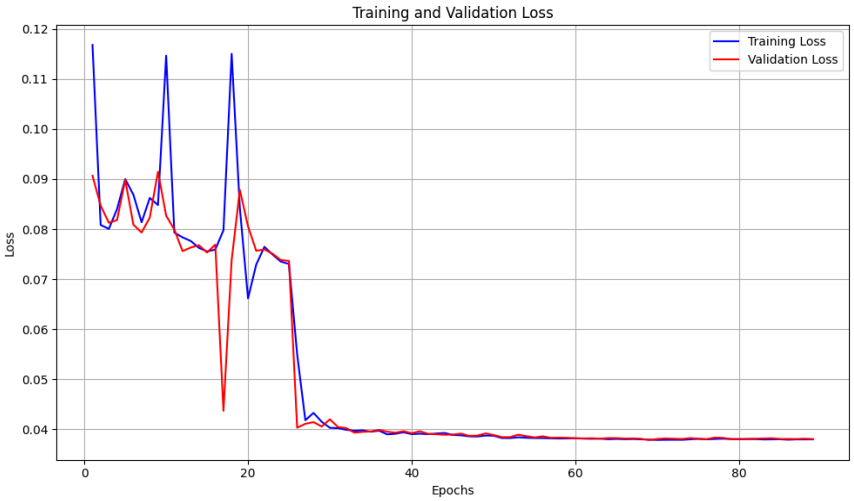
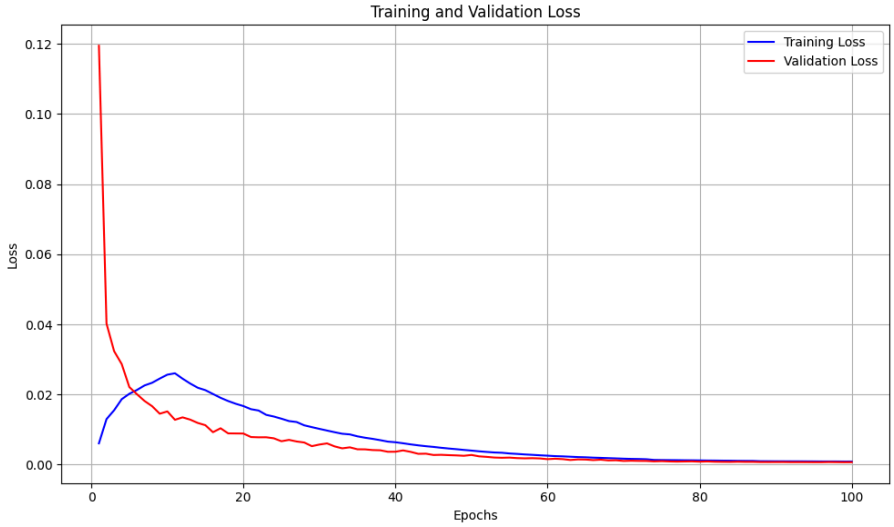


Figure 8: Upsampling loss plot - Final version



7.3 Future Work

Improve the WaveGAN and DiffWave models by testing different hyperparameters such as batch sizes, learning rates, number of epochs, and loss functions to further improve the quality of the generated audio. Use the same methodology to train the models to generate drum samples other than snare drums. Then, we can try to generate audio of different instruments such as piano and guitar.

Team Contribution

Preprocessing the drum sample audio - Dalton

Modifying and training WaveGAN - Jun, Abhijeet

Modifying and training DiffWave - Abhijeet, Dalton

Implementing and training Custom upsampling model - Dalton, Jun, Abhijeet

Implementing IPM - Jun, Abhijjeet

Audio Comparison Metries (LSD and MCD) - Dalton, Jun

Testing the model and generating the samples - Dalton, Jun, Abhijjeet

References

Bińkowski, M., Sutherland, D. J., Arbel, M., & Gretton, A. (2018). Demystifying mmd gans. arXiv preprint arXiv:1801.01401. https://arxiv.org/abs/1801.01401

Donahue, C., McAuley, J., & Puckette, M. (2018). Adversarial audio synthesis. arXiv preprint arXiv:1802.04208. https://arxiv.org/abs/1802.04208

Kong, Z., Ping, W., Huang, J., Zhao, K., & Catanzaro, B. (2020). Diffwave: A versatile diffusion model for audio synthesis. arXiv preprint arXiv:2009.09761. https://arxiv.org/abs/2009.09761

Sharvil N. "diffwave." GitHub.https://github.com/lmnt-com/diffwave.

.

Mostafa E. “WaveGAN” https://github.com/mostafaelaraby/wavegan-pytorch.