<Final Report: 4팀>

Guitar Amp Emulation with Neural Style Transfer

Donghoon Hyun
School of Electrical and Electronic Engineering
College of Engineering
Yonsei University

<Final Report: 4팀>

Guitar Amp Emulation with Neural Style Transfer

Report Advisor: Hong-gu Kang

December 2024

Donghoon Hyun
School of Electrical and Electronic Engineering
College of Engineering
Yonsei University

Contents

ABSTRACT

Numerous attempts have been made to emulate the response of guitar amplifiers and distortion effects. However, due to their inherent nonlinearity, conventional DSP-based approaches have proven to be inefficient. This project leverages neural networks to effectively emulate distortion effects, including the manipulation on timbre based on user control input, similar to how traditional effectors function. The proposed model utilizes mel spectrograms with a U-Net as the backbone network and BigVGAN as the vocoder. The model is implemented in Python/Pytorch and available on GitHub. The model generates audio according to the training target and user control input, even under previously unseen conditions, within a reasonable timeframe. While the quality may not yet reach the level of state-of-the-art models, the system effectively captures the influence of user control input on timbre.

Key words : Neural Net, Style Transfer, Guitar Amp, Distortion, Mel-Spectrogram, U-Net, BigVGAN

1. Introduction

Guitarists often rely on specialized equipment called effectors and guitar amplifiers, not only to amplify the sound but also to shape the timbre, creating the signature sound of the electric guitar. These devices are essential for achieving iconic tones, yet many guitarists find them prohibitively expensive. To address the demand for more affordable solutions, there have been numerous efforts to emulate the responses of guitar amplifiers and effectors using digital signal processing (DSP). However, since the signal from the guitar is processed through electric circuits consisting of various components such as vacuum tubes and transistors, the responses are nonlinear, making DSP-based approach challenging.

Neural networks, known for their ability to model complex, unknown functions, offer a potential solution. In this work, I propose a pipeline for emulating the response of a distortion effector using neural networks. The method manipulates the mel spectrogram of the input signal with a U-Net, conditioned by a FiLM layer generated from user conditioning inputs. With this approach, guitarists can avoid the high costs of physical equipment by simply downloading the model and enjoying their play. The implementation and sound samples are available on GitHub[1].

2. Related Work

2.1 Mel Spectrogram

Thanks to the Short Time Fourier Transform (STFT), an audio signal can be represented both in the time and frequency domains. This method operates by splitting the waveform into small time segments and then applying the Fast Fourier Transform (FFT) to each segment. It shows the frequency components presenting in short time intervals, enabling more precise analysis of the signal's characteristics.

However, from the perspective of human perception, a linear frequency representation

[1] ‘https://github.com/dhun222/Guitar-amp-emulator-with-neural-style-transfer’

in Hz is not efficient, because the human ear perceives frequency information on an exponential scale. Mel spectrogram exploits the fact by scaling, dividing, and compressing the frequency range to address the problem. The mel spectrogram is obtained by passing the linear spectrogram through a special filter called the "mel filter." This filter consists of multiple "mel bins" that partition the frequency range into several bands. Each frequency range is then scaled and compressed into the corresponding mel bin. As a result, some frequency information is lost during this process since multiple frequencies are represented within a single mel bin. Despite the lossy nature of this transformation, mel spectrograms are widely used in many audio processing models. This is because they effectively reduce the size of the data while maintaining most of the perceptually relevant information.

2.2 U-Net

U-Net is a convolutional neural network originally designed for image segmentation. Due to its flexible architecture, it has been widely used for various general image processing tasks, such as image reconstruction and pansharpening. Its name is derived from the U-like shape formed by its architecture, which consists of three main components: the encoder, the mid-block and the decoder. These components are connected serially and residually to process images.

The encoder progressively downsamples the input image using max pooling. During this process, the width and height of the feature map reduces, while the number of channels increases. The encoder also includes convolution layers connected to downsampling layers to catch the spatial characteristics of feature map.

After passing through the encoder, the feature map proceeds to the mid-block, which consists of additional convolution and activation layers. In the proposed model of this work, FiLM layers are incorporated into the mid-block for user input conditioning. A detailed concept will be discussed in a later section of the paper.

The decoder is similar to the revert of the encoder, upsampling the feature map using

transposed convolutions or interpolation. During this upsampling process, the width and height of the feature map increase, while the number of channels decreases, essentially reversing the operations of the encoder. This structure helps the output image to recover the original shape of the input image. However, residual connections between the downsampling and upsampling layers double the number of channels after each upsampling operation, differing the decoder from the exact inversion of the encoder. This enables the model to retain and utilize high resolution information when reconstructing the image from low resolution.

In the proposed model of this work, the sound is processed in the form of mel spectrogram which is in 2-D shape, enabling the U-Net to handle the sound data which is originally 1-D shaped signal.

2.3 FiLM

Feature-w**i**se **L**inear **M**odulation (FiLM) is a general conditioning technique for CNN, initially designed to enhance visual reasoning tasks. This method operates by manipulating the features in the activation map of CNN layers through affine transformations applied in a feature-wise manner. It requires only two parameters per modulated feature. It's computationally effective since the number of parameters remains independent of the input data's resolution and requires only a small amount of parameters. These parameters are generated with input conditioning. In the original paper, conditioning is derived from question for visual reasoning, and GRU generates the corresponding parameters. In this work, the parameters are generated based on user control, which represents the configuration of the target effector. Further details will be provided in the following sections.

2.4 BigVGAN

As mentioned earlier, many neural network models leverage mel spectrograms to process or generate audio data. However, since some information is lost during the transformation into a mel spectrogram, reconstructing it back into a waveform requires

additional processing. The neural network model designed for this task is called a "vocoder."

BigVGAN is a universal vocoder based on the Generative Adversarial Network (GAN) architecture. GAN consists of two kinds of networks: the generator and the discriminator. The generator generates data, while the discriminator distinguishes between real and generated data. Often, multiple discriminators are employed for more accurate training. The generator is trained to fool the discriminator(s), while the discriminator(s) are trained to separate between real and generated data in a competitive manner.

Previous GAN-based vocoders were designed for specific tasks, such as speech generation, and typically perform poorly when applied to domains outside their training data. In contrast, BigVGAN is a general purposed vocoder capable of generating sound across a variety of domains, including music, speech, etc. It overcomes the limitations of previous models with its large parameters and newly introduced discriminator. Additionally, it incorporates a periodic activation function and anti-aliased representations within the generator to improve the performance.

Generator

The generator is composed of several stacks of transposed convolution layers and an Anti-aliased Multi-Periodicity composition (AMP) module. Each layer progressively upsamples the resolution of the mel spectrogram. The AMP module utilizes a periodic activation function to capture multiple periodicities and the inductive bias inherent in the audio signal.

Discriminator

The discriminator employs two discriminators, each consisting of multiple sub-discriminators. It borrows Multi-Periodic Discriminator (MPD) from HiFi-GAN, while also introducing a new discriminator called Multi-Resolution Discriminator (MRD). The MPD reshapes the 1-D waveform signal into a 2-D representation with varying widths and heights to separately capture multiple periodic structures with multiple sub-discriminators using 2-D convolutions. The MRD consists of several sub-discriminators that operate on linear spectrograms at different STFT resolutions.

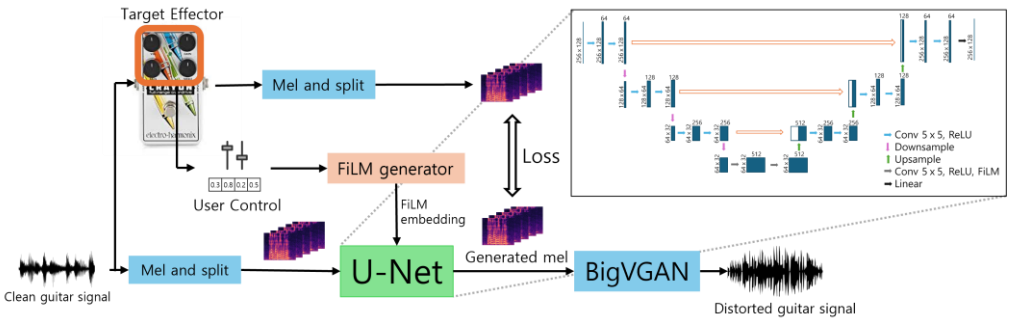

Fig 1. The pipeline of proposed model including training

Loss Functions

BigVGAN uses several loss functions for both the generator and the discriminators. For the generator, the loss function is a combination of adversarial loss, feature matching loss and the L1 norm distance between the generated mel spectrogram and the real data. The adversarial loss measures the difference between the discriminator's judgment and the ground truth. The feature matching loss measures the difference between the features generated by the sub-discriminators in each layer and the features of the ground truth. For the discriminators, the training objective is the sum of adversarial loss and feature matching loss.

3. Proposed Model

3.1 Architecture

The U-Net architecture is limited to accept only 2D inputs of fixed size, which presents an obstacle when applying it directly to audio signals. To address this issue, the data preprocessing module converts the audio signal into a mel spectrogram, then divides it into same sized segments. The mel spectrogram effectively represents the audio signal in 2D shape (time and frequency). Moreover, this segmentation allows the serial pieces to be batched together, enabling efficient processing on GPUs.

Each downsampling and upsampling layer in the U-Net comprises two consecutive

convolutional and activation layers, being followd by a max pooling or transposed convolution layer. Residual connections are incorporated at each depth level, concatenating the output feature map of the downsampling layers to that of the upsampling layers along the channel axis which leads to a doubling of the channel count in the resulting feature map.

The mid-block of the U-Net follows a similar structure to a single block of upsampling/downsampling layer, with two consecutive convolutional and activation layers. After the activation layer, the FiLM layer modulates the activation maps allowing user input to control the manipulation of the audio's timbre. The FiLM generator processes user control input, which represents the user's control on the target effector, and generates the corresponding parameters for FiLM layer. This generator consists of a single learnable fully connected layer, with output dimensions matching the number of channels in the modulated activation maps. A detailed configuration of the U-Net with ablation is provided in the experiment section.

The resulting output by U-Net is passed to the BigVGAN, which synthesizes the corresponding waveform. The model leverages pretrained parameters, using the same audio and mel spectrogram configuration as in the preprocessing module.

3.2 Training and Inference Pipeline

The model is trained using supervised learning. During the training, input data is provided to both the model and the target effector, and the loss is computed based on the difference between the model's output as prediction and the target effector's output as ground truth. Specifically, the loss is calculated between the mel spectrogram generated by the U-Net and the mel spectrogram of the ground truth.

User control over the target effector is also included into the training. The changes in user control influence the ground truth. Simultaneously, the levels of each control knob on the effector are fed to the FiLM generator of the model. The U-Net, conditioned by the FiLM layer, generates a mel spectrogram that corresponds to the given user control input. As the ground truth reflects the effects of these control changes, the model learns to understand and replicate

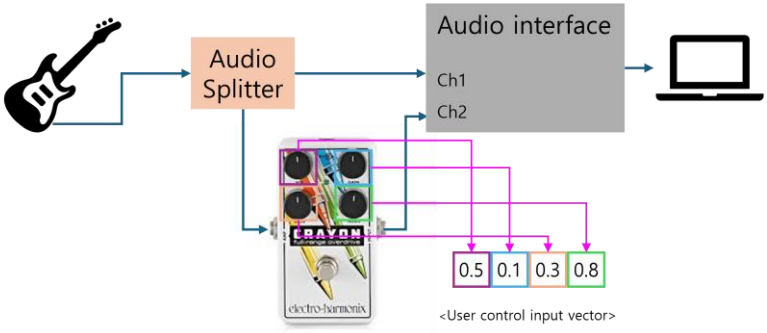

Fig 2. Methodology for data collection

the influence of the user input on the generated output.

During the inference, the target effector and loss function are omitted. To use the model, the user puts the user control input in the same form as the target effector and the raw sound of electric guitar. Then the model generates the sound replacing a role of target effector which only modifies the timbre while preserving other elements of the sound such as notes, rhythm etc.

3.3 Loss Function

The loss function is applied to the mel spectrogram which is generated by U-Net directly. The loss function is as follows,

$$Loss = \mathbb{E}(\left|U\big(P(x)\big) - P(y)\right|_{L1})$$

where x denotes the input signal, y denotes the ground truth obtained from target effector's output, P denotes the data preprocessing module and U denotes the U-Net.

4. Experiment

4.1 Dataset

Since the training process employs supervised learning, the model requires pairs of input and corresponding ground truth data. To obtain these pairs, I recorded both the raw input and the output of the target effector using an audio splitter. This approach enabled the precise

acquisition of input-output pairs from the target effector. The training dataset consists of two types of data: real guitar play and white noise. The real guitar data is expected to capture some inductive bias and reflect the distribution of real-world data, while the white noise dataset contains a broad range of frequencies, which is intended to efficiently reflect the target effector's response across the frequency domain. Both datasets were recorded under identical configurations, though they differ in length.

Table 1. Configurations and specifications of training and test set

Training data	**# samples**	**Total length**	**Sample rate**	**Bit depth**
Real play White noise	27	2.7h 13.5m	44.1kHz	16bit
Test data	**# samples**	**Total length**	**Sample rate**	**Bit depth**
Real play	6	3m	44.1kHz	16bit

User control inputs for the target effector were also collected. Audio data was recorded corresponding to the user control inputs. The target effector has four knobs: volume, gain, bass, and treble. The settings of each knob were measured and normalized to floating-point values between 0 and 1. Consequently, the target effector provides four floating-point numbers corresponding to the four knobs, forming a four-dimensional user control input vector. During the recording process, the volume knob was fixed at 50%, and each of the other knobs was manipulated at three levels: 20%, 50%, and 80%. This results in 27 unique combinations of user control inputs and corresponding audio input-output data pairs.

Additionally, a test dataset was designed, consisting of six pairs of real play data which differ from the training data, conditioned on user control inputs which were also not seen during training. Table 1 provides a detailed description of the configurations and specifications of the datasets.

4.2 Ablation

To investigate the contribution of model size to the performance, an experiment was

conducted with two different sizes. The two models differ in their depth of downsampling and upsamping layers. Table 2 provides details of the different configuration of U-Net based on their size.

Table 2. Configurations of U-Net in two different sizes

Model size	# parameters	Depth	Channels
Large	10M	3	[64, 128, 256]
Small	3.7M	2	[64, 128]

4.3 Evaluation Metric

Table 3 describes the evaluation metrics that are used to measure the performance of the model. The subscript $_W$ denotes that the metric is applied to the waveform output by BigVGAN while subscript $_{mel}$ denotes that the metric is applied to the generated mel spectrogram by U-Net directly. Hence the three formal metrics with the subscript $_W$ are applied to the generated waveform and $L1_{mel}$ with the subscript $_{mel}$ is applied to the generated mel spectrogram by U-Net.

Table 3. Description of evaluation metrics

Metric	**Description**
LSD_W	Distance of log-scaled spectrograms between prediction and GT
ESR_W	Error-to-signal ratio of prediction and GT
$MSTFT_W$	Average distance with diverse STFT resolutions
$L1_{mel}$	L1 distance between prediction and GT

5. Result

5.1 Audio Quality

To measure the quality of generated waveform, the evaluation metrics introduced above are used. Table 4 provides the results of the experiment. These results indicate that increasing the model size does not significantly improve performance. In fact, for some metrics, such as

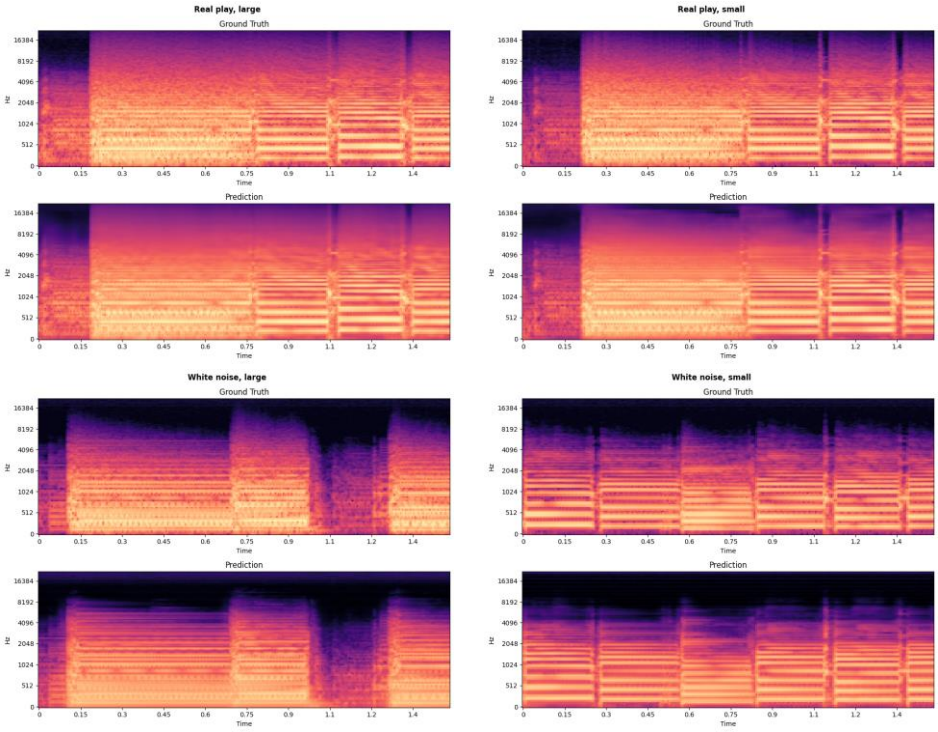

Fig 3. Mel spectrogram of prediction and ground truth of each configuration and data

$MSTFT_W$ and $L1_{mel}$, the smaller model outperforms the larger model. Additionally, it is observed that the model trained with real play data achieves better performance than the model trained with white noise data.

Table 4. Result of the experiments

Training Data	Model size	$LSD_W \downarrow$	$ESR_W \downarrow$	$MSTFT_W \downarrow$	$L1_{mel} \downarrow$
Real play	Large	**0.8939**	**1.9162**	1.3097	0.6119
	Small	0.9006	1.9606	**1.2763**	**0.5786**
White noise	Large	1.5878	4.7344	2.8740	0.9356
	Small	1.3645	4.1371	2.4437	0.8313

Figure 3 presents the mel spectrograms of the prediction and ground truth. The overall structure, reflecting elements such as volume, notes, and rhythm, is nearly identical. However, notable differences are observed in the texture of the spectrograms, which represent the

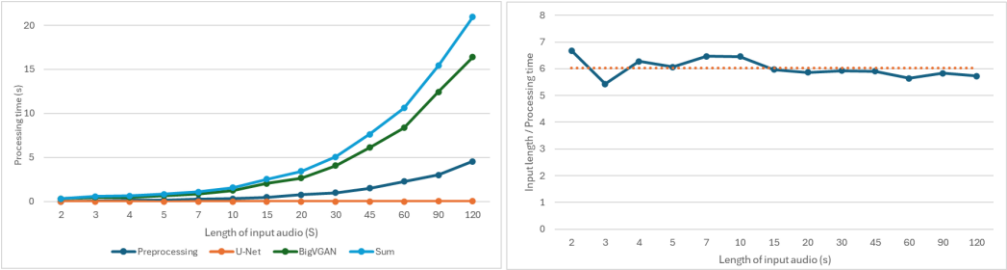

Fig 4. Processing time of the model with large configuration

harmonics and timbre of the sound. The ground truth exhibits a more detailed, fine-grained pattern, while the prediction displays smoother grains along the time axis, indicating a loss of nuance and resulting in a difference in timbre.

5.2 Processing Speed

Time spent on the inference of the model is also measured. Figure 4 shows the processing time of the model with large configurations. Each step in the inference pipeline against the length of input data is represented separately. BigVGAN takes the largest portion of processing time while U-Net takes smallest portion. Figure _ shows the ratio of processing time and length of input data. The average value is 6.02, which means that processing time is 1/6.02 of input length and is reasonable value for real time processing.

6. Conclusion

In this work, several neural network models are employed to emulate the response of a distortion effector. The proposed model successfully learns the tendencies of user control for the target effector and generates output in a reasonable amount of time. It appears that the model performs better when trained with real play data rather than artificial data. Although the quality does not yet reach state-of-the-art levels, the model demonstrates the potential to replace conventional distortion effectors in the perspective of shaping timbre while preserving other

elements of the sound.

However, although the model effectively reflects user control in generation, it has limitations that it can only learn a single target effector at a time. Additionally, the data collection process is time-consuming and not user-friendly, which could be challenging for individuals unfamiliar with the proposed pipeline. Since the primary objective of this research is to enhance the convenience for guitar players, further exploration to overcome these limitations is necessary.

Referrence

[1] EthanPerez, FlorianStrub, HarmdeVries, VincentDumoulin, & AaronCourville. (2017). FiLM: Visual Reasoning with a General Conditioning Layer.“AAAI.”

[2] KongJungil, KimJaehyeon, & BaeJaekyoung. (2020). HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis.“NeurIPS.”

[3] Lee, S.-g., Ping, W., Ginsburg, B., Catanzaro, B., & Yoon, S. (2022). BigVGAN: A Universal Neural Vocoder with Large-Scale Training. *ICLR 2023.*

[4] “Librosa Documentation.” https://librosa.org/doc/main/index.html

[5] LiuHaohe, ChenZehua, YuanYi, MeiXinhao, LiuXubo, MandicDanilo, . . . PlumbleyD.Mark. (2023). AudioLDM: Text-to-Audio Generation with Latent Diffusion Models.“ICML.”

[6] Oordvan denAaron, DielemanSander, ZenHeiga, SimonyanKaren, VinyalsOriol, GravesAlex, . . . KavukcuogluKoray. (2016). WaveNet: A Generative Model for Raw Audio.

[7] “Pytorch Documentation.” https://pytorch.org/docs/stable/index.html

[8] RonnebergerOlaf, FischerPhilpp, & BroxThomas. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation.“MICCAI.”

[9] WriteAlec, DamskaggEero-Pekka, JuvelaLauri, & ValimakiVesa. (2019). Real-Time Guitar Amplifier Emulation with Deep Learning.“SMC.”