Deep Neural Network Modeling of Distortion Stomp Box Using Spectral Features

Kento Yoshimoto, Hiroki Kuroda, Daichi Kitahara, and Akira Hirabayashi
Graduate School of Information Science and Engineering, Ritsumeikan University, Shiga, Japan
E-mail: is0383ip@ed.ritsumei.ac.jp, {kuroda, d-kita, akirahrb}@media.ritsumei.ac.jp

***Abstract*—We propose a distortion stomp box modeling method using a deep neural network. A state-of-the-art method exploits a feedforward variant of the original autoregressive WaveNet. The modified WaveNet is trained so as to minimize a loss function defined by the normalized mean squared error between the high-pass filtered outputs. This method works well for stomp boxes with low distortion, but not for those with high distortion. To solve this problem, we propose a method using the same WaveNet, but a new loss function, which is defined by a weighted sum of errors in the time and frequency domains. The error in the time domain is the mean squared error without high-pass filtering. The error in the frequency domain is the generalized Kullback–Leibler (KL) divergence between spectrograms, which are given with a short-time Fourier transform (STFT) and a Mel filter bank. Numerical experiments using a stomp box with high distortion, the Ibanez SD9, show that the proposed method is capable of reproducing high-quality sounds compared with the state-of-the-art method especially for high-frequency components.**

I. Introduction

When one plays an electric guitar, the sound is sometimes distorted using a stomp box or amplifier. Players carefully choose particular ones from many different types of stomp boxes and amplifiers to create their intended tone. Note that certain sounds can be produced only by ones so-called "vintage" or ones whose production have already finished. Those products are in great demand and hard to obtain. Digital modeling to reproduce the sound of such devices is one good alternative to meet the demand.

Digital modeling techniques can be classified into two types. Methods of the first type convert all electronic circuits in the devices into mathematical models [1], [2]. This circuit-based approach is capable of generating high-quality results. Such modelings, however, require not only the circuit diagram of the target device but also characteristic curves of all nonlinear circuit parts including transistors, diodes, and vacuum tubes. Even worse, if the circuit diagram is not available, huge efforts for reverse engineering are required.

Methods of the second type are machine learning [3]–[12]. Using pairs of clean input and distorted output sounds of the target device, the mapping from the input to output is learned. Cost to collect such data is much lower than that for the circuit-based approach, as long as the target device is available.

Methods of the second type are further classified into two subgroups. Those in the first subgroup are called *block-oriented models* and use a little information about the electric circuits in the device, which typically consist of linear filtering blocks followed by a nonlinear block. Thus, the block-oriented models use the same block structure. The pairs of the clean input and distorted output sounds of the target device are used to adjust the parameters in the blocks [3]–[7].

On the other hand, methods in the second subgroup use no information about the electric circuits in the device. Instead, these methods exploit deep learning. Since guitar sounds are temporal sequences, recurrent neural networks (RNNs) are fit for the modeling of the stomp boxes. Indeed, long short-term memory (LSTM) networks [13], which is one of the RNNs, are used in [8] and [9]. These methods are capable of modeling with high accuracy. It takes, however, a long time to train the LSTM networks due to their recursive structures.

For faster training, Damskägg *et al.* proposed a modeling method based on a feedforward variant of WaveNet [10]–[12], which was originally proposed to synthesize audio waveforms, including human voice and music, using a nonlinear autoregressive structure [14]. WaveNet does not use recursive structures and hence training is fast. Further, the so-called *dilated causal convolution* enables WaveNet to reproduce high-quality sounds with low computational cost. The modified WaveNet is trained so as to minimize the error-to-signal ratio (ESR) loss function, defined by the normalized mean squared error between high-pass filtered target and modeling sounds in the time domain. This method achieved better results with faster training than the LSTM methods. Nevertheless, low-frequency components are attenuated by the side effect of the high-pass filter and the reproduction of the high-frequency components is not enough. Thus, stomp boxes with high distortion are not well modeled by this method.

To solve the above problems, we propose a novel modeling method, in which the same modified WaveNet is used as in [11]. On the other hand, the loss function is differently defined by a weighted sum of errors in the time and frequency domains. The error in the time domain is the mean squared error without high-pass filtering to avoid the attenuation of the low-frequency components. The error in the frequency domain is defined by the generalized Kullback–Leibler (KL) divergence between spectrograms of target and modeling sounds. As the spectrograms, we use the Mel-frequency power spectrograms defined by the squared absolute values of a short-time Fourier transform (STFT) followed by a Mel filter bank. Numerical experiments using a high distortion stomp box, the Ibanez SD9, show that the proposed method reproduces the high-frequency components well without the attenuation of the low-frequency components compared with the conventional method [11].

978-988-14768-8-3/20/$31.00 ©2020 APSIPA

II. Preliminaries

A. Neural Network Model for Distortion Stomp Box

Let $x[n]$ and $y[n]$ respectively be the input and the output signals of the distortion stomp box at discrete time instant $n \in \{1, 2, \ldots, N\}$. As a black box model of the distortion stomp box, we adopt a state-of-the-art deep neural network (DNN) model based on WaveNet [10]–[12]. WaveNet is originally proposed in [14] as an autoregressive model which predicts a future sample from past samples, and is modified in [10]–[12] as a feedforward model which computes an output signal from input signals.

The overall structure of the neural network used in this paper is shown in Fig. 1. At time n, the network $f_{\boldsymbol{\vartheta}}$ computes the modeling sound $\hat{y}[n]$ from R input signals, i.e.,

$$\hat{y}[n] = f_{\boldsymbol{\vartheta}}(x[n-R+1], x[n-R+2], \ldots, x[n]), \quad (1)$$

where $\boldsymbol{\vartheta}$ denotes the parameters of the network to be trained so that the target sound $y[n]$ is approximated well by $\hat{y}[n]$. Note that R is set to a large enough value so that $f_{\boldsymbol{\vartheta}}$ can approximate well the characteristic of the distortion stomp box.

The detail of each layer of the network is as follows. The pre-processing layer converts the single-channel input signal $x[n]$ to an L channel signal as

$$\boldsymbol{x}_0[n] = \boldsymbol{w}_0 x[n] + \boldsymbol{b}_0, \quad (2)$$

where $\boldsymbol{w}_0 \in \mathbb{R}^L$ represents a convolutional filter, $\boldsymbol{b}_0 \in \mathbb{R}^L$ is a bias term.

Then, the L channel signal $\boldsymbol{x}_0[n]$ passes K residual blocks, which are connected in a sequence. For $k = 1, 2, \ldots, K$, the kth residual block computes two outputs $\boldsymbol{x}_k[n]$ and $\boldsymbol{s}_k[n]$ from the input $\boldsymbol{x}_{k-1}[n]$. To this end, each residual block first computes two dilated causal (DC) convolutions as

$$\begin{cases} \boldsymbol{u}_{k,1}[n] = \displaystyle\sum_{m=0}^{M} \boldsymbol{W}_{k,1}[m]\boldsymbol{x}_{k-1}[n - md_k] + \boldsymbol{b}_{k,1}, \\ \boldsymbol{u}_{k,2}[n] = \displaystyle\sum_{m=0}^{M} \boldsymbol{W}_{k,2}[m]\boldsymbol{x}_{k-1}[n - md_k] + \boldsymbol{b}_{k,2}, \end{cases} \quad (3)$$

where $\boldsymbol{W}_{k,1}[m], \boldsymbol{W}_{k,2}[m] \in \mathbb{R}^{L \times L}$ $(m = 0, 1, \ldots, M)$ are convolutional filters of size $M+1$, d_k is the dilation factor, and $\boldsymbol{b}_{k,1}, \boldsymbol{b}_{k,2} \in \mathbb{R}^L$ are bias terms. DC convolution is employed to enlarge the number R of input signals used by the network while maintaining low computational complexity (see Fig. 2 for an illustration). The value of the dilation factor is doubled as the layer progresses, and is reset to 1 when exceeds 256, i.e., $(d_1, d_2, \ldots, d_9, d_{10}, d_{11}, \ldots) = (1, 2, \ldots, 256, 1, 2, \ldots)$. Since R is given by $R = M\left(\sum_{k=1}^{K} d_k\right) + 1$, we can enlarge the number R of inputs while keeping the filter size $M+1$ small.

After the DC convolutional layer, the gated activation unit computes

$$\boldsymbol{v}_k[n] = g\left(\boldsymbol{u}_{k,1}[n]\right) \odot g\left(\boldsymbol{u}_{k,2}[n]\right), \quad (4)$$

where $\odot$ denotes the component-wise multiplication, and g is the component-wise soft-sign activation function $\frac{u}{1+|u|}$. The output $\boldsymbol{x}_k[n]$ of the kth residual block is obtained by mixing

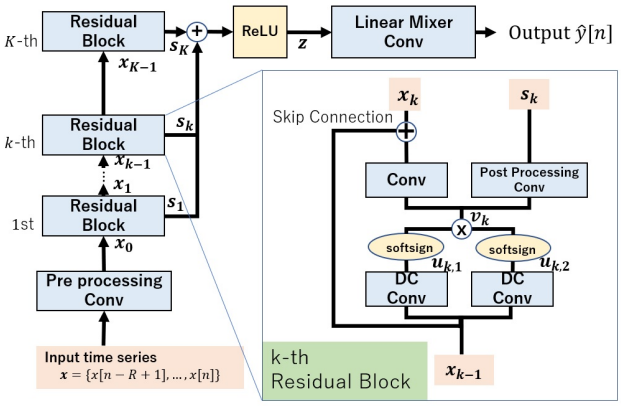


Fig. 1. Neural network model for distortion stomp box.

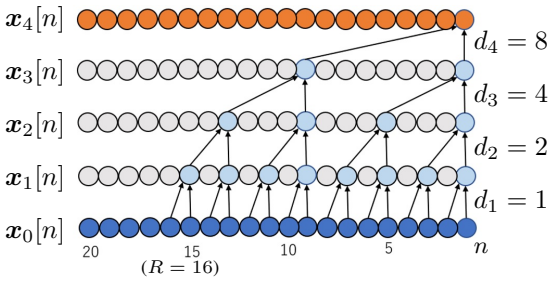


Fig. 2. Dilated causal convolution ($K = 4$ and $M = 1$).

the output of the activation unit and the input of this block:

$$\boldsymbol{x}_k[n] = \boldsymbol{W}_{k,3}\boldsymbol{v}_k[n] + \boldsymbol{b}_{k,3} + \boldsymbol{x}_{k-1}[n], \quad (5)$$

where $\boldsymbol{W}_{k,3} \in \mathbb{R}^{L \times L}$ is a convolutional filter and $\boldsymbol{b}_{k,3} \in \mathbb{R}^L$ is a bias term. The other output $\boldsymbol{s}_k[n]$ is computed as

$$\boldsymbol{s}_k[n] = \boldsymbol{W}_{k,4}\boldsymbol{v}_k[n] + \boldsymbol{b}_{k,4}, \quad (6)$$

where $\boldsymbol{W}_{k,4} \in \mathbb{R}^{L \times L}$ and $\boldsymbol{b}_{k,4} \in \mathbb{R}^L$.

The outputs $\boldsymbol{s}_1[n], \boldsymbol{s}_2[n], \ldots, \boldsymbol{s}_K[n]$ of the residual blocks are merged by the so-called skip connections, and then processed through the rectified linear unit (ReLU):

$$\boldsymbol{z}[n] = \mathrm{ReLU}\left(\sum_{k=1}^{K} \boldsymbol{s}_k[n]\right), \quad (7)$$

where ReLU computes $\max(0, s)$ component-wisely. Finally, a single channel modeling sound $\hat{y}[n]$ is obtained by applying a convolution:

$$\hat{y}[n] = \boldsymbol{w}_{K+1}^{\top}\boldsymbol{z}[n], \quad (8)$$

where $\boldsymbol{w}_{K+1} \in \mathbb{R}^L$. Note that learnable parameters of the network are $\boldsymbol{w}_0$, $\boldsymbol{b}_0$, $\boldsymbol{W}_{k,j}$, $\boldsymbol{b}_{k,j}$ $(k = 1, 2, \ldots, K; j = 1, 2, 3, 4)$, and $\boldsymbol{w}_{K+1}$. Totally, the network has $2K\{L^2(M+2) + 2L\} - L^2 + 2L$ parameters. In general, the expressiveness of the network is enhanced by increasing the number of parameters, at the cost of the computational complexity. In the modeling of the distortion stomp boxes, low latency is very important, and thus fewer parameters are preferable.

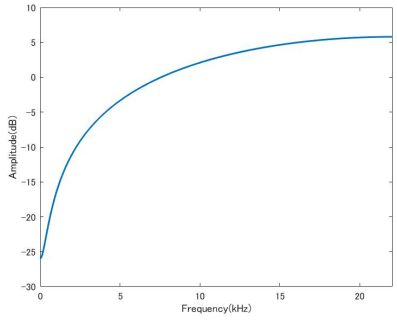


Fig. 3. Frequency response of $H(z)$ (sampling frequency 44.1 kHz).

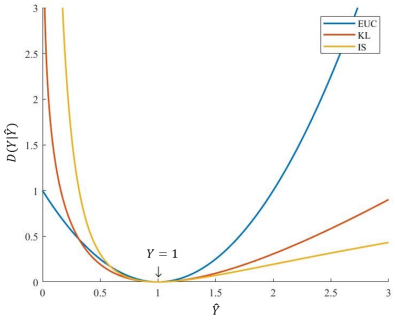


Fig. 4. Three error measures for spectrograms.

B. Existing Strategy for Network Training

To improve the modeling quality for high-frequency components, the existing methods [10]–[12] train the network with high-pass filtered target and output sounds. More precisely, the network is trained by minimizing the error-to-signal ratio (ESR)

$$\mathrm{ESR} = \frac{\sum_{n=1}^{N} \left(\hat{y}_{\mathrm{f}}[n] - y_{\mathrm{f}}[n]\right)^2}{\sum_{n=1}^{N} y_{\mathrm{f}}[n]^2}, \tag{9}$$

where $\hat{y}_{\mathrm{f}}[n]$ and $y_{\mathrm{f}}[n]$ are filtered modeling sounds and target sounds with a high-pass filter $H(z) = 1 - 0.95z^{-1}$ (see Fig. 3 on the frequency response of $H(z)$). Because of the high-pass filtering, the network trained by this strategy tends to disregard low-frequency components. In addition, simply applying the high-pass filtering is still insufficient for precise modeling of the high-frequency components (see Table II in Section IV for the modeling quality for the high-frequency components).

III. Proposed Method

A. Design of Spectral Features

To reproduce the high-frequency components more faithfully without sacrificing the accuracy of the low-frequency components, we propose to combine the frequency-domain error with the time-domain error. As spectral features, we use the power spectrogram (PS) and the Mel-frequency power spectrogram (MFS). The proposed methods using PS and MFS are referred to as Method PS and Method MFS, respectively.

The power spectrograms $\boldsymbol{Y}_{\mathrm{pow}} \in \mathbb{R}_+^{I\times J}$ and $\hat{\boldsymbol{Y}}_{\mathrm{pow}} \in \mathbb{R}_+^{I\times J}$ are respectively computed from the target sound $y[n]$ and the modeling sound $\hat{y}[n]$ by a short-time Fourier transform (STFT), where $\mathbb{R}_+$ denotes the set of all nonnegative real numbers, I is the number of frequency bins, and J is the number of frames. We also utilize the Mel-frequency power spectrograms $\boldsymbol{Y}_{\mathrm{mel}} \in \mathbb{R}_+^{\tilde{I}\times J}$ and $\hat{\boldsymbol{Y}}_{\mathrm{mel}} \in \mathbb{R}_+^{\tilde{I}\times J}$ computed by applying a Mel filter bank to the power spectrograms $\boldsymbol{Y}_{\mathrm{pow}}$ and $\hat{\boldsymbol{Y}}_{\mathrm{pow}}$, respectively, where $\tilde{I}$ is the number of Mel-frequency bins.

B. Proposed Loss Function

The error between two spectrograms $\boldsymbol{Y} = (Y_{i,j})$ and $\hat{\boldsymbol{Y}} = (\hat{Y}_{i,j})$ is normally measured by the Euclidean distance, generalized Kullback–Leibler (KL) divergence, and Itakura–Saito (IS) divergence, which are defined by

$$\mathrm{EUC}(\boldsymbol{Y} \,\|\, \hat{\boldsymbol{Y}}) = \sum_{i=1}^{I}\sum_{j=1}^{J} \left(\hat{Y}_{i,j} - Y_{i,j}\right)^2, \tag{10}$$

$$\mathrm{KL}(\boldsymbol{Y} \,\|\, \hat{\boldsymbol{Y}}) = \sum_{i=1}^{I}\sum_{j=1}^{J} \left(Y_{i,j} \log \frac{Y_{i,j}}{\hat{Y}_{i,j}} - (Y_{i,j} - \hat{Y}_{i,j})\right), \tag{11}$$

and

$$\mathrm{IS}(\boldsymbol{Y} \,\|\, \hat{\boldsymbol{Y}}) = \sum_{i=1}^{I}\sum_{j=1}^{J} \left(\frac{Y_{i,j}}{\hat{Y}_{i,j}} - \log \frac{Y_{i,j}}{\hat{Y}_{i,j}} - 1\right), \tag{12}$$

respectively. For the case of $I = J = 1$, the values of these functions in terms of $\hat{Y} > 0$ with $Y = 1$ are shown in Fig. 4. The Euclidean distance is symmetric with respect to $Y = 1$. On the other hand, the generalized KL and IS divergences are asymmetric and penalize more for $\hat{Y} < 0.3$ and less for $\hat{Y} > 1$ than the Euclidean distance. From our experience, the modified WaveNet [11] tends to have small output values. Thus, the two asymmetric lower-more-penalizing divergences are expected to train the network more accurately than the symmetric Euclidean distance. Based on the results shown in Tables I to III in Section IV, we adopt the generalized KL divergence for evaluation of the spectrograms, because it showed the best performance in average. In contrast to our expectation, the IS divergence performed worse than the Euclidean distance.

Based on these observations, the proposed method uses the mean squared error without the high-pass filter for the time-domain loss function l_{time}, and the generalized KL divergence for the frequency-domain loss function l_{freq}. More precisely, we define the loss function for the network $f_{\boldsymbol{\vartheta}}$ by

$$\mathrm{Loss}(\boldsymbol{\vartheta}) = l_{\mathrm{time}}(\boldsymbol{\vartheta}) + \lambda l_{\mathrm{freq}}(\boldsymbol{\vartheta}), \tag{13}$$

where

$$l_{\mathrm{time}}(\boldsymbol{\vartheta}) = \frac{1}{N}\sum_{n=1}^{N} (\hat{y}[n] - y[n])^2, \tag{14}$$

and

$$l_{\mathrm{freq}}(\boldsymbol{\vartheta}) = \frac{1}{IJ}\mathrm{KL}(\boldsymbol{Y} \,\|\, \hat{\boldsymbol{Y}}). \tag{15}$$

Note that, for the sake of simplicity, we omit the dependency on $\boldsymbol{\vartheta}$ in the notations of $\hat{y}[n]$ and $\hat{Y}_{i,j}$. The parameter $\lambda > 0$

controls the relative importance of the time-domain waveform and the spectral features. In the loss function, the frequency-domain term $l_{\mathrm{freq}}(\boldsymbol{\vartheta})$ evaluates only the power of the spectrogram, but not the phase, while the time-domain term $l_{\mathrm{time}}(\boldsymbol{\vartheta})$ compensates it. Taking the balance between them by λ, the proposed loss function accurately evaluates the error between the network outputs and the target sounds. In the experiments shown in Section IV, we use $\lambda = 1$ and $\lambda = 0.1$ respectively for the cases of the power spectrogram $\boldsymbol{Y} = \boldsymbol{Y}_{\mathrm{pow}}$ and the Mel-frequency power spectrogram $\boldsymbol{Y} = \boldsymbol{Y}_{\mathrm{mel}}$.

The details of the spectral features used in the proposed method are as follows. From the target sounds $y[1], y[2], \ldots, y[N]$, we compute the target power spectrogram $\boldsymbol{Y}_{\mathrm{pow}} = (Y_{i,j}^{\mathrm{pow}}) \in \mathbb{R}_+^{I \times J}$ with the STFT as

$$Y_{i,j}^{\mathrm{pow}} = \left| \sum_{n=1}^{N_{\mathrm{f}}} \psi[n] y[(j-1)\tau + n] e^{-2\sqrt{-1}\pi \frac{(i-1)(n-1)}{N_{\mathrm{f}}}} \right|^2, \quad (16)$$

for $i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, J$, where $\psi[n]$ is a window function, τ is the frame shift, and N_{f} is the frame length. Note that the number of frequencies is given as $I = \left\lceil \frac{N_{\mathrm{f}}+1}{2} \right\rceil$ and the number of frames is given as $J = \left\lceil \frac{N}{\tau} \right\rceil$, where $\lceil \cdot \rceil$ is the ceiling function, and zero padding is used for the outside parts $y[N+1], y[N+2], \ldots, y[(J-1)\tau + N_{\mathrm{f}}]$. The modeling power spectrogram $\hat{\boldsymbol{Y}}_{\mathrm{pow}}$ is also computed from the modeling sounds $\hat{y}[1], \hat{y}[2], \ldots, \hat{y}[N]$ in the same way. Next, the target Mel-frequency power spectrogram $\boldsymbol{Y}_{\mathrm{mel}} = (Y_{b,j}^{\mathrm{mel}}) \in \mathbb{R}_+^{\tilde{I} \times J}$ is computed from $\boldsymbol{Y}_{\mathrm{pow}}$ by using a Mel filter bank as

$$Y_{b,j}^{\mathrm{mel}} = \sum_{i=1}^{I} H_b[i] Y_{i,j}^{\mathrm{pow}}, \quad (17)$$

for $b = 1, 2, \ldots, \tilde{I}$ and $j = 1, 2, \ldots, J$, where $H_b[i]$ is the Mel-scale band-pass filter [15, Section 6.5.2]. In the same way, the modeling Mel-frequency power spectrogram $\hat{\boldsymbol{Y}}_{\mathrm{mel}}$ is computed from $\hat{\boldsymbol{Y}}_{\mathrm{pow}}$ with the same Mel filter bank. In the following simulations, we use the frequency band from 60 Hz to 22 kHz, which is divided into $\tilde{I} = 300$ bins.

IV. Numerical Experiments

A. Experimental Setup

We used a high distortion stomp box, the Ibanez SD9, in this experiment. The Ibanez SD9 has three knobs, each of which corresponds to distortion, tone, and volume. They were set to the direction of 12 o'clock, or the middle position.

The modified WaveNet structure as set as follows: channel number $L = 16$, residual block number $K = 18$, and filter size $M + 1 = 3$, thus $M = 2$. Hence, we have $2K\{L^2(M+2) + 2L\} - L^2 + 2L = 37{,}792$ adjustable parameters. Further, $d_k = 2^{k-1}$ for $1 \le k \le 9$, $d_k = 2^{k-10}$ for $10 \le k \le 18$, and $R = M\left(\sum_{k=1}^{K} d_k\right) + 1 = 2{,}045$, which approximately corresponds to 46.4 ms when the sampling frequency is 44.1 kHz.

The training process was implemented with Keras in Python 3.7.3. The computational environment is Windows 10 Pro, Core i9-7980X, 128 GB main memory, GeForce GTX1080Ti GPU.

B. Training and Test Data

For training data, we exploited the IDMT dataset [16], [17], where the sampling frequency is 44.1 kHz and the bit depth is 16 bits. A total of 5 minutes of data (150 seconds of guitar sounds and 150 seconds of bass sounds) were randomly selected from the dataset. They were used as clean input signals and sent to the stomp box through a reamper (Radial ProRMP [20]) to generate the corresponding distorted output signals. These data were trimmed at every 100 ms so that $D = 3{,}000$ pairs $\{\boldsymbol{t}_{\mathrm{train}}^{(1)}, \boldsymbol{t}_{\mathrm{train}}^{(2)}, \ldots, \boldsymbol{t}_{\mathrm{train}}^{(D)}\}$ of $N = 4{,}410$ input and output sequences were obtained. Each pair $\boldsymbol{t}_{\mathrm{train}}^{(d)}$ is decomposed into N elements $\{t_{\mathrm{train}}^{(d,1)}, t_{\mathrm{train}}^{(d,2)}, \ldots, t_{\mathrm{train}}^{(d,N)}\}$, where $t_{\mathrm{train}}^{(d,n)}$ consists of a single output value $y[n]$ and R input values $x[n-R+1]$, $x[n-R+2], \ldots, x[n]$, from which $\hat{y}[n]$ is computed. For $n < R$, zeros were filled into $x[n-R+1], x[n-R+2], \ldots, x[0]$.

For each training pair $\boldsymbol{t}_{\mathrm{train}}^{(d)}$, we compute the spectral features used in the proposed loss function as follows. From the target sounds $y[1], y[2], \ldots, y[N]$ in $\boldsymbol{t}_{\mathrm{train}}^{(d)}$, we compute the target power spectrogram $\boldsymbol{Y}_{\mathrm{pow}} \in \mathbb{R}_+^{I \times J}$ as (16), where we set the frame shift to $\tau = 256$, and the frame length to $N_{\mathrm{f}} = 1{,}024$. Note that this setting implies that the number I of frequency bins is 513 and the number J of frames is 18. In (16), we use the hann window

$$\psi[n] = \frac{1}{2} - \frac{1}{2} \cos\left(2\pi \frac{n-1}{N_{\mathrm{f}} - 1}\right) \quad (n = 1, 2, \ldots, N_{\mathrm{f}}). \quad (18)$$

Similarly, we obtain the modeling power spectrogram $\hat{\boldsymbol{Y}}_{\mathrm{pow}}$ from the modeling sound $\hat{y}[1], \hat{y}[2], \ldots, \hat{y}[N]$ computed from the input values in $\boldsymbol{t}_{\mathrm{train}}^{(d)}$. The Mel-frequency power spectrograms $\boldsymbol{Y}_{\mathrm{pow}}$ and $\hat{\boldsymbol{Y}}_{\mathrm{mel}}$ are computed as (17).

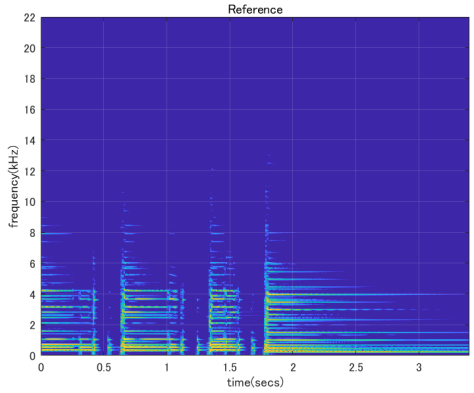
Since the amount of data is huge ($DN \approx 1.3 \times 10^7$), it is difficult to compute the value and the gradient of the loss function for the overall training data $\{\boldsymbol{t}_{\mathrm{train}}^{(1)}, \boldsymbol{t}_{\mathrm{train}}^{(2)}, \ldots, \boldsymbol{t}_{\mathrm{train}}^{(D)}\}$. Thus, we utilize the so-called mini-batch training. The overall training data is randomly divided into $P = 16$ mini-batch data $\{\boldsymbol{t}_{\mathrm{train}}^{(d_p[1])}, \boldsymbol{t}_{\mathrm{train}}^{(d_p[2])}, \ldots, \boldsymbol{t}_{\mathrm{train}}^{(d_p[D_p])}\}$ ($p = 1, 2, \ldots, P$). The value and the gradient of the loss function are computed for each mini-batch data. We say that an "epoch" is completed when all of P mini-batch data are used for training. An iterative optimization algorithm, Adam [18], is applied to minimize the loss function. The iteration is repeated for 1,000 epochs, or until an early stopping condition is met, where the condition is evaluated by other two 300 pairs of guitar and bass signals randomly selected from the IDMT dataset.

To evaluate the trained networks, we prepared original four sound sources.[1] In sound source 1, a $\mathrm{B}\flat_{\mathrm{add9}}$ chord was played with high attacks. In sound source 2, a $\mathrm{D}_{\mathrm{sus2}}$ chord was played with low attacks. In sound source 3, a chromatic scale from E2 to A♭3 was played with high attacks. In sound source 4, two tones of D3 and D4 were played with low attacks.

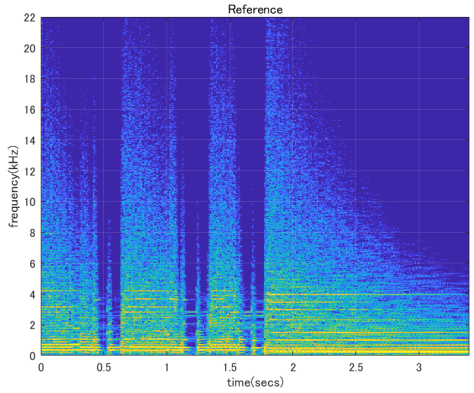
C. Experimental Results

Figure 5 shows simulation results for sound source 1. Figures (a) and (b) are the power spectrograms of the clean input

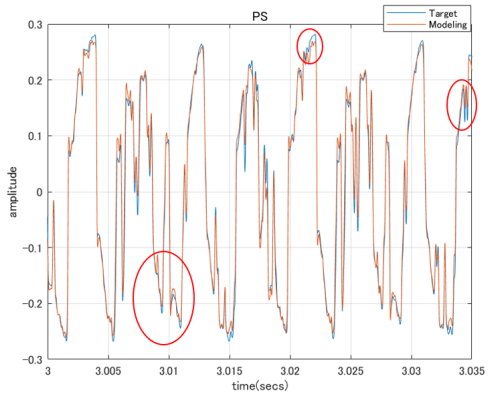
[1] You have access to the sound sources from our web site [19].



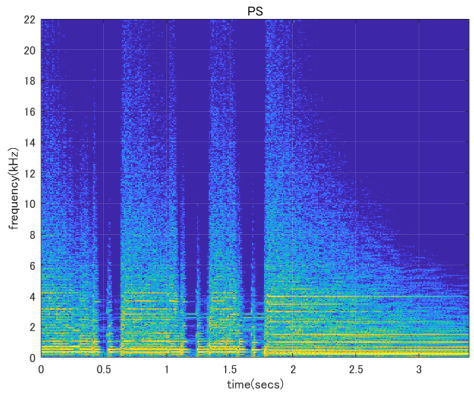
(a) Power spectrogram of the input sound $x[n]$



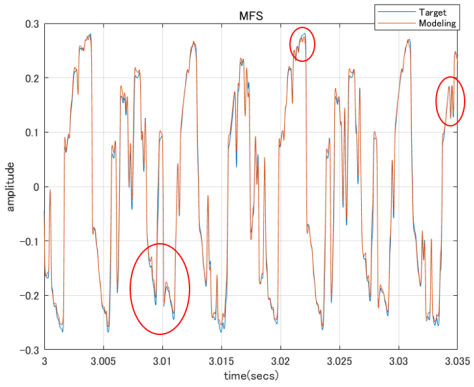
(b) Power spectrogram of the target sound $y[n]$



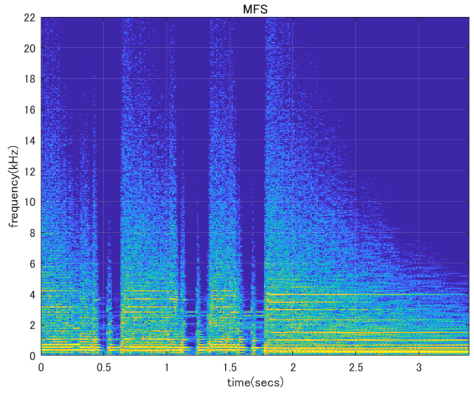
(c) Waveform generated by Method PS-KL



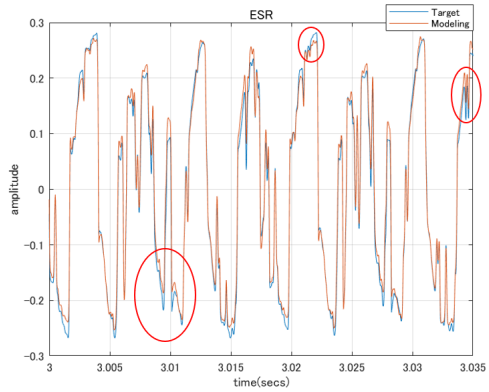
(d) Power spectrogram of the waveform in (c)



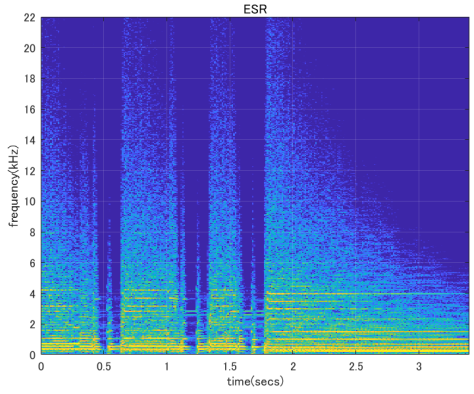
(e) Waveform generated by Method MFS-KL



(f) Power spectrogram of the waveform in (e)



(g) Waveform generated by the conventional method in [11]



(h) Power spectrograms of the waveform in (g)

Fig. 5. Simulation results for the Ibanez SD9 with sound source 1.

TABLE I
ESR WITHOUT THE HIGH-PASS FILTER.

Method	Sound 1	Sound 2	Sound 3	Sound 4	average
PS-EUC	2.71%	0.79%	0.77%	0.59%	1.21%
PS-KL	1.41%	**0.60%**	0.67%	**0.38%**	0.76%
PS-IS	11.71%	7.93%	6.20%	5.17%	5.77%
MFS-EUC	2.57%	0.90%	0.76%	0.74%	1.24%
MFS-KL	**1.21%**	**0.60%**	**0.53%**	0.45%	**0.70%**
MFS-IS	2.76%	1.25%	1.67%	0.91%	1.65%
Conv. [11]	1.62%	0.69%	0.95%	0.55%	0.95%

TABLE II
ESR WITH THE HIGH-PASS FILTER.

Method	Sound 1	Sound 2	Sound 3	Sound 4	average
PS-EUC	17.69%	9.99%	7.09%	13.25%	12.01%
PS-KL	11.68%	7.36%	5.19%	9.86%	8.52%
PS-IS	50.49%	50.45%	39.32%	54.99%	48.81%
MFS-EUC	17.93%	8.41%	6.23%	12.23%	11.20%
MFS-KL	**9.01%**	**4.51%**	**2.57%**	**6.65%**	**5.69%**
MFS-IS	22.38%	14.42%	9.90%	18.57%	16.31%
Conv. [11]	16.86%	7.29%	5.03%	7.15%	9.08%

TABLE III
NMSE OF THE POWER SPECTROGRAM.

Method	Sound 1	Sound 2	Sound 3	Sound 4	average
PS-EUC	0.40%	0.20%	0.35%	**0.09%**	0.26%
PS-KL	0.33%	0.17%	0.36%	0.10%	0.24%
PS-IS	3.32%	2.93%	5.82%	2.87%	3.74%
MFS-EUC	0.48%	0.25%	3.33%	0.15%	0.30%
MFS-KL	**0.28%**	**0.16%**	**0.26%**	0.10%	**0.20%**
MFS-IS	0.92%	0.58%	1.63%	0.33%	0.87%
Conv. [11]	0.77%	0.50%	1.01%	0.42%	0.68%

sound and the distorted target sound. The modeling waveform generated by Method PS-KL is indicated in Figure (c) by a red line with the target waveform indicated by a blue line. Figure (d) shows the corresponding power spectrogram. Figure (e) shows the modeling waveform generated by Method MFS-KL with red as well as the target waveform with blue. Figure (f) shows the corresponding power spectrogram. Figures (g) and (h) indicates the modeling waveform generated by the conventional method in [11] and the corresponding power spectrogram, respectively. By comparing the parts indicated by the circles in Figs. (c), (e), and (g), we can see that the proposed methods, Methods PS-KL and MFS-KL, reproduced the target sound more accurately than the method in [11].

To compare these results objectively, we computed the ESR in (9) for each method and each sound source. Each value computed not using the high-pass filter is shown in Table I. We can see that Methods PS-KL and MFS-KL improved the ESR by 20.0% and 26.3%, respectively, in average. ESR computed using the high-pass filter is shown in Table II. We can see that Methods PS-KL and MFS-KL improved the ESR by 6.2% and 37.3%, respectively, in average. It is interesting that our methods outperformed the method in [11] in the sense of the ESR with the high-pass filter, which is the loss function for the conventional method. It is difficult to see the difference of the spectrograms in Figs. (d), (f), and (h). Nevertheless, we can clarify the difference from the normalized mean squared error (NMSE) shown in Table III, which indicates that Methods PS-KL and MFS-KL improved NMSE by 64.7% and 70.6%. These results mean that the high-frequency components are reproduced more precisely using the frequency-domain evaluation than using a high-pass filter. Finally, it was shown that the proposed loss function works more effectively with the Mel-frequency power spectrogram than with the power spectrogram. Thus, in conclusion, Method MFS-KL performed the best among the methods compared in this paper.

V. CONCLUSIONS

We proposed a modeling method for stomp boxes with high distortion using the modified WaveNet. We exploited the same network structure as the conventional method [11]. To train the network, we proposed a new loss function, which is defined by a weighted sum of errors in the time and frequency domains. The error in time domain is the mean squared error without high-pass filtering. The error in the frequency domain is the generalized Kullback–Leibler (KL) divergence between Mel-frequency power spectrograms of target and modeling sounds. Numerical experiments using a stomp box with high distortion, the Ibanez SD9, showed that the proposed method can reproduce high-quality sounds more than the conventional method especially for high-frequency components.

REFERENCES

[1] D. T. Yeh, J. Abel, and J. O. Smith, "Simulation of the diode limiter in guitar distortion circuits by numerical solution of ordinary differential equations," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Bordeaux, France, Sep. 2007, pp. 197–203.

[2] D. T. Yeh and J. O. Smith, "Simulating guitar distortion circuits using wave digital and nonlinear state-space formulations," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Espoo, Finland, Sep. 2008, pp. 19–26.

[3] A. Novak, L. Simon, P. Lotton, and J. Gilbert, "Chebyshev model and synchronized swept sine method in nonlinear audio effect modeling," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Graz, Austria, Sep. 2010, 4 pages.

[4] R. C. D. de Paiva, J. Pakarinen, and V. Välimäki, "Reduced-complexity modeling of high-order nonlinear audio systems using swept-sine and principal component analysis," in *Proc. AES Int. Conf. Appl. Time-Freq. Process. Audio*, Helsinki, Finland, Mar. 2012, 10 pages.

[5] F. Eichas and U. Zölzer, "Black-box modeling of distortion circuits with block-oriented models," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Brno, Czech Republic, Sep. 2016, pp. 39–45.

[6] F. Eichas, S. Möller, and U. Zölzer, "Block-oriented gray box modeling of guitar amplifiers," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Edinburgh, UK, Sep. 2017, pp. 184–191.

[7] F. Eichas, *System Identification of Nonlinear Audio Circuits*. Helmut Schmidt University, Ph.D. thesis, Oct. 2019.

[8] Z. Zhang, E. Olbrych, J. Bruchalski, T. J. McCormick, and D. L. Livingston, "A vacuum-tube guitar amplifier model using long/short-term memory networks," in *Proc. IEEE SoutheastCon*, St. Petersburg, FL, USA, Apr. 2018, 5 pages.

[9] Y. Matsunaga, N. Aoki, Y. Dobashi, and T. Yamamoto, "A digital modeling technique for distortion effect based on a machine learning approach," in *Proc. Asia-Pac. Signal Inf. Process. Assoc. Annual Summit Conf. (APSIPA ASC)*, Honolulu, HI, USA, Nov. 2018, pp. 1888–1892.

[10] E.-P. Damskägg, L. Juvela, and V. Välimäki, "Deep learning for tube amplifier emulation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Brighton, UK, May 2019, pp. 471–475.

[11] E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time modeling of audio distortion circuits with deep learning," in *Proc. Sound Music Comput. Conf. (SMC)*, Málaga, Spain, May 2019, pp. 332–339.

[12] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time guitar amplifier emulation with deep learning," *Appl. Sci.*, vol. 10, no. 3, 18 pages, Jan. 2020.
[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[14] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv:1609.03499*, 15 page, Sep. 2016.
[15] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Upper Saddle River, NJ: Prentice Hall, 2001.
[16] https://www.idmt.fraunhofer.de/en/business_units/m2d/smt/guitar.html, Jan. 20, 2020.
[17] https://www.idmt.fraunhofer.de/en/business_units/m2d/smt/bass_lines.html, Jan. 20, 2020.
[18] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, 15 pages.
[19] http://www.ms.is.ritsumei.ac.jp/SoundSource_fx_eng.html, June 19, 2020.
[20] https://www.radialeng.com/product/prormp, June 29, 2020.