# ANTI-ALIASING OF NEURAL DISTORTION EFFECTS VIA MODEL FINE TUNING

*Alistair Carson* , Alec Wright and Stefan Bilbao*

Acoustics and Audio Group
University of Edinburgh
Edinburgh, UK
<alistair.carson, alec.wright, s.bilbao>@ed.ac.uk

## ABSTRACT

Neural networks have become ubiquitous with guitar distortion effects modelling in recent years. Despite their ability to yield perceptually convincing models, they are susceptible to frequency aliasing when driven by high frequency and high gain inputs. Nonlinear activation functions create both the desired harmonic distortion and unwanted aliasing distortion as the bandwidth of the signal is expanded beyond the Nyquist frequency. Here, we present a method for reducing aliasing in neural models via a teacher-student fine tuning approach, where the teacher is a pre-trained model with its weights frozen, and the student is a copy of this with learnable parameters. The student is fine-tuned against an aliasing-free dataset generated by passing sinusoids through the original model and removing non-harmonic components from the output spectra. Our results show that this method significantly suppresses aliasing for both long-short-term-memory networks (LSTM) and temporal convolutional networks (TCN). In the majority of our case studies, the reduction in aliasing was greater than that achieved by two times oversampling. One side-effect of the proposed method is that harmonic distortion components are also affected. This adverse effect was found to be model-dependent, with the LSTM models giving the best balance between anti-aliasing and preserving the perceived similarity to an analog reference device.

## 1. INTRODUCTION

Systems for nonlinear waveshaping, filtering and amplification are essential tools in music production and performance, in particular electric guitar playing. Devices such as vacuum tube amplifiers and transistor-based fuzz pedals have been used since the 1960s to shape the sound of the electric guitar through the introduction of harmonic distortion to the spectrum. In the digital domain, the simplest distortion effects can be implemented with clipping or saturating non-linear functions. Most digital distortion effects, however, seek to model or emulate analog devices [1]. Methods for virtual analog modelling of distortion effects include circuit-based white-box methods [2, 3], and black-box modelling including neural network approaches [4, 5, 6, 7].

An inherent problem with non-linear digital audio processing is aliasing distortion, which is often perceived as unpleasant artefacts, beating or noise [8]. The harmonics generated by deliberate clipping of a signal often exceed the Nyquist frequency, therefore causing aliasing within the audio band. The canonical method for reducing aliasing is oversampling – processing at rates of two or more times the audio rate (see e.g. [9]). Alternative methods have also been explored, but are often limited to a certain class or subset of functions, e.g. bandlimited interpolation applied to soft-clipping functions [10]. Parker et al. [11] proposed a method based on continuous-time convolution of the distorted signal with an anti-aliasing low-pass filter, known as antiderivative anti-aliasing (ADAA). This method and variants thereof work well for memoryless nonlinear functions [11, 12, 13, 14] and virtual analog models with one or two states [15, 16, 17]. However, the application of ADAA to larger state-space systems or neural networks is an open research question.

In general, anti-aliasing in the context of neural distortion effects constitutes an interesting research problem. Vanhatalo et al. [18] considered various methods: using (synthetic) oversampled training data; low-pass filter placement between layers; incorporating spectral loss functions into training; and using sparse networks through model pruning. Out of these, forced sparsity was found to be a viable option for a temporal convolutional network (TCN), but was less effective for the long-short-term-memory network (LSTM) example. Furthermore, it came at the cost of reduced model accuracy [18]. Köper and Holters [19] proposed an anti-aliased state-trajectory network model, and whilst in some cases a reduction in aliasing was shown, the main limitation was that the model could only be trained on synthetic data, not audio from an analog device. Our previous work [20, 21] showed that $M$ times oversampling can be implemented in LSTMs by adjusting the feedback delay length to $M$ samples. With an appropriate design of interpolation and decimation filters [22], this can be employed to reduce aliasing, but of course comes at the expense of $M$ times more operations per input sample.

Here, we investigate a data-driven approach to reducing the aliasing caused by neural network models of distortion effects without any modifications to the model architecture itself or oversampling. This paper is structured as follows: Sec. 2 outlines the proposed methodology; Sec. 3 describes the casy study models; Sec. 4 presents objective results and spectral analysis; Sec. 5 contains a perceptual evaluation; and Sec. 6 provides concluding remarks. Open source code and audio examples are available [1].

## 2. METHODOLOGY

Consider an audio processing neural network of the form:

$$y = f(x, \theta) \tag{1}$$

* A. Carson is funded by the Scottish Graduate School of Arts and Humanities (SGSAH)



[1] https://a-carson.github.io/dafx25_antialiasing_neural/

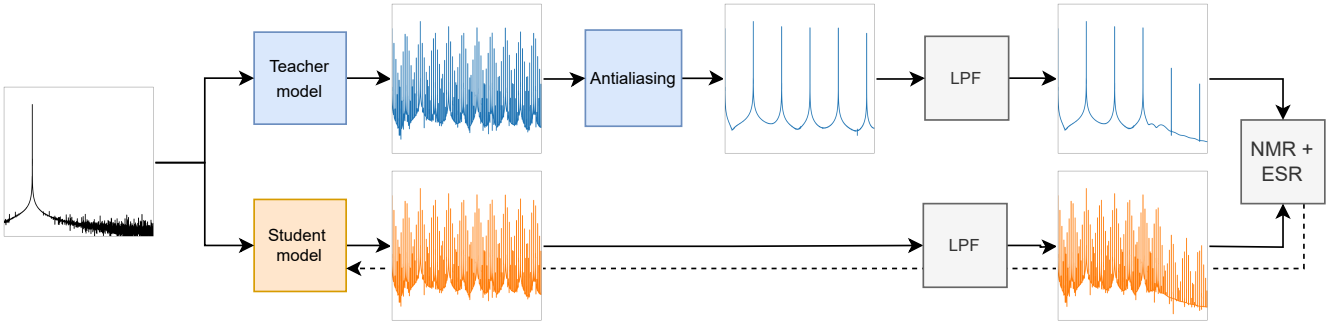



Figure 1: *Fine-tuning procedure for anti-aliasing of the Student model. The dashed line indicates the flow of gradients to the Student parameters. Spectral plots are included for illustration only – training operates in the time domain (except for the NMR calculation).*

where $x$ is the input signal, $y$ is the output signal and $\theta$ are the model parameters or weights. Our methodology is not model-specific, so we allow $f$ to be any neural network including but not limited to recurrent neural networks (RNNs) and TCNs. Despite the fact that RNNs and TCNs can both yield perceptually convincing models of distortion effects [6, 23], it has been shown that the non-linear activation functions, while responsible for creating the desired harmonic distortion effect, also generate aliasing [18, 24]. We assume that $f$ has been pre-trained against some ground truth audio output from an analog device, $y_{\text{ref}}$, but that we no longer necessarily have access to the original device or the training data collected from it. For example, the model may have been obtained as open-weight from elsewhere (e.g. [25]). The first step in our method is to duplicate the pre-trained model, with the original version designated as the *teacher model* and the other the *student model*. Formally these are defined:

$$y_{\text{teach}} = f(x, \theta_{\text{teach}}) \tag{2a}$$

$$y_{\text{stud}} = f(x, \theta_{\text{stud}}), \tag{2b}$$

where initially $\theta_{\text{stud}} = \theta_{\text{teach}} = \theta$. The weights of the teacher model are then frozen and randomised sine tones are passed through the model to produce a synthetic dataset of signals which can be decomposed into harmonics plus noise (including but not limited to aliasing noise). We assume that the "perfect" output signal in virtual analog modelling is purely harmonic, so remove all non-harmonic components from the teacher model output using spectral analysis and Fourier synthesis. These aliasing-free signals are then used as a target during the training of the student model weights. This "fine-tuning" procedure is illustrated in Fig. 1 and details are covered in the remainder of this section.

### 2.1. Input sine tones

The input to both the teacher and student model during training is a batch of pseudo-randomly generated sine tones, defined as:

$$x[n] = A \sin(2\pi f_0 n / F_s + \phi), \tag{3}$$

where $F_s$ is the sampling rate (and set to 44.1 kHz unless otherwise specified). The amplitude $A$ and phase $\phi$ are sampled from uniform distributions:

$$A \sim \mathcal{U}(0, 1), \quad \phi \sim \mathcal{U}(0, 2\pi) \tag{4}$$

and the frequency is set according to:

$$f_0 = 440 \cdot 2^{(l-69)/12}, \quad l \sim \mathcal{U}(21, 127). \tag{5}$$

The bounds on $l$ were chosen as the range of midi notes on a standard keyboard, but here $l$ may be integer or non-integer valued. The sinusoids are processed through the teacher model to generate the signal $y_{\text{teach}}$ which contains a mixture of harmonic distortion and aliasing noise components.

### 2.2. Aliasing removal

The teacher model output is then post-processed to remove aliasing. First the initial samples are truncated, removing transients to leave a one-second segment. The signal is then multiplied by a Chebyshev window $w[n]$ with 120 dB side-lobe attenuation and the FFT taken to obtain the spectrum:

$$Y_{\text{teach}}[k] = \sum_{n=0}^{N-1} w[n] y_{\text{teach}}[n] e^{-j2\pi k/N}. \tag{6}$$

Given a sinusoidal input, the harmonic frequencies of the model output occur at, for integer $m$:

$$f_m = (m+1) f_0 \tag{7}$$

where $0 \leq m \leq M-1$, and $M = \lfloor F_s/(2f_0) \rfloor$. The closest FFT bins to the harmonic frequencies are therefore:

$$b_m = \text{round}(f_m N / F_s) \tag{8}$$

and the difference between these bins to the "true" bin indices are:

$$d_m = f_m N / F_s - b_m. \tag{9}$$

Now the complex amplitudes of the harmonic components can be extracted from the spectrum and adjusted for the scalloping loss that arises when the harmonic is an off-bin frequency [26, 27]:

$$C_m = \frac{Y_{\text{teach}}[b_m]}{\sum_{n=0}^{N-1} w[n] e^{j2\pi n d_m / N}}. \tag{10}$$

Finally, a bandlimited alias-free version of the signal is constructed using Fourier synthesis:

$$\tilde{y}_{\text{teach}}[n] = B + 2 \sum_{m=0}^{M-1} |C_m| \cos\left(2\pi f_m n + \angle C_m\right) \tag{11}$$

where $B$ is the DC offset (adjusted for the windowing gain):

$$B = \frac{Y_{\text{teach}}[0]}{\sum_{n=0}^{N-1} w[n]}. \tag{12}$$

From here on, the tilde notation denotes the bandlimited aliasing-free version of a distorted sinusoidal signal.

### 2.3. Pre-emphasis filtering

Pre-emphasis filtering is a common technique in black-box modelling of audio effects [23, 28]. Prior to loss computation in our model, both the target $\tilde{y}_{\text{teach}}$ and student model output $y_{\text{stud}}$ are filtered by a low-pass filter (LPF) with passband and stopband edges of 12 kHz and 16 kHz respectively and 80 dB stopband attenuation. The justification for this is that aliasing is much more noticeable when it appears below the fundamental frequency of the desired signal and human hearing depreciates considerably above 16 kHz [8]. The filter therefore encourages very high frequency errors to be ignored during training, and initial experiments showed that this improved results. A comparison against other pre-emphasis filters was considered but not included in the scope of this work.

### 2.4. Loss function

The loss between the student model output and the alias-free teacher output is computed as the sum of the error-to-signal ratio (ESR) and the noise-to-mask ratio (NMR). The ESR is commonly used in black-box modelling of audio effects [23, 5, 6]. For any arbitrary signal $\hat{s}$ and a reference signal $s$, the ESR is defined as:

$$\mathbb{E}(\hat{s}, s) = \frac{\sum_{n=0}^{N-1}(s[n] - \hat{s}[n])^2}{\sum_{n=0}^{N-1} s[n]^2}. \tag{13}$$

Furthermore, in this work we use the noise-to-mask ratio (NMR) in the loss function and in evaluation. The NMR is the energy ratio between non-harmonic components and the simplified masking threshold of desired harmonic components [29, 30] and has been used in several works as a perceptually-informed aliasing measurement[8, 9, 14, 27], and as a loss function in neural watermarking [31]. The NMR is computed between STFTs of the signal and reference signal, so let us denote the STFT of signal $s$ as $S_{k,t}$ for $k = 0, \ldots, K-1$, $t = 0, \ldots, T-1$ where $K$ and $T$ are the number of frequency bins and time frames respectively. For an FFT length of $N_{\text{FFT}}$ this gives $K = N_{\text{FFT}}/2 + 1$ bins.

The first step in NMR calculation is to compute the STFTs of signal $\hat{s}$ and the reference $s$ to obtain $\hat{S}_{k,t}$ and $S_{k,t}$ respectively. The *noise pattern* is then computed:

$$N_{c,t} = \sum_{k=0}^{K-1} U_{c,k} \Big( \omega_k (|\hat{S}_{k,t}| - |S_{k,t}|) \Big)^2 \tag{14}$$

where $\omega_k$ is a filter approximation of the human inner and outer ear; and $U_{c,k}$ are elements of a $C \times K$ matrix that maps the STFT frequencies from $K$ FFT bins to $C$ critical bands (the rows are bandpass filters for each band). The *masking pattern* is computed from the reference STFT as:

$$M_{c,t} = \mathcal{S}\left( \sum_{k=0}^{K-1} U_{c,k} \cdot |\omega_k \cdot S_{k,t}|^2 + \mu_c \right) \tag{15}$$

where $\mu_c$ is the internal ear noise for each band and $\mathcal{S}(\cdot)$ is a function that spreads energy between the critical bands to account for frequency masking – details of which are omitted here for brevity but the reader is referred to the work of Kabal (Section 2.8) [30]. The noise-to-mask ratio is then computed:

$$\text{NMR}(\hat{S}, S) = \frac{1}{C \cdot T} \sum_{c=0}^{C-1} \sum_{t=0}^{T-1} \frac{N_{c,t}}{M_{c,t}} \tag{16}$$

For this work we adapt the MATLAB implementation provided by Zheleznov [27] into a differentiable PyTorch module, available in the accompanying code. A window and FFT size of $N_{\text{FFT}} = 2048$ was used with an overlap of 50% and $C = 109$ critical bands.

Given the student model output $y_{\text{stud}}$, the bandlimited teacher model output $\tilde{y}_{\text{teach}}$ and their respective STFTs $S_{\text{stud}}$ and $\tilde{S}_{\text{teach}}$, the loss function used in fine tuning is therefore:

$$\mathcal{L} = \mathbb{E}(y_{\text{stud}}, \tilde{y}_{\text{teach}}) + \lambda \cdot \text{NMR}(S_{\text{stud}}, \tilde{S}_{\text{teach}}). \tag{17}$$

Both terms in the loss function penalise aliasing in the student model output as well as spectral differences between the harmonic distortion components of the student and teacher models. Here we set $\lambda = 1$ with a study on the effect of $\lambda$ left for further work.

### 2.5. Training details

In each training batch the input is a set of sine tones of duration 1.2 seconds. These are processed through the teacher model, truncated to 1 second ($N = F_s$ samples) and anti-aliased to obtain the target batch. For RNN-based models, the first $N/5$ samples were processed through the student model to initialise the states, then truncated back-propagation through time (TBPTT) was implemented with a frame size of $N/10$ samples. For TCN models, the first $N/5$ samples were discarded and the rest processed in frames of $N/2$ samples (due to memory constraints only). The batch size was 40 for RNN models and 32 for TCN models. All models were trained using Adam optimizer with a learning rate of 5e-4 for a maximum of 40k iterations (maximum 24 hours on a NVIDIA GeForce GTX 1080). Double precision was used.

### 2.6. Validation dataset and metrics

The models were validated and tested using two datasets: an audio signal comprised of 60s of guitar and bass recordings; and a set of sine tones "playing" the chromatic scale from midi note number 21 to 108 ($f_0$ ranging from of 27.5 Hz to 4186 Hz). Each tone was repeated at three different amplitudes: $-36$ dB, $-18$ dB and $-6$ dB. The signals were passed through the teacher model, student model and reference analog device (if available) to obtain the corresponding $y_{\text{teach}}$, $y_{\text{stud}}$ and $y_{\text{ref}}$. The guitar and bass (G+B) data and the sine tone data are then analysed separately.

On the G+B data we compute the ESR and a multi-resolution spectral convergence loss (MRSL) between log-magnitude mel-spectrograms of the signal and reference [32, 33]. Where the reference analog device was not available, the teacher model output was used as the reference. For each sine tone input the following metrics are computed on output signal $y$ (either $y_{\text{teach}}$ or $y_{\text{stud}}$):

**ESR-R**: the ESR w.r.t. the reference, $\mathbb{E}(y, y_{\text{ref}})$;
**NMR-R**: the NMR w.r.t. the reference, $\text{NMR}(S, S_{\text{ref}})$;
**HESR-R**: the ESR of the magnitude of the harmonic components w.r.t. the reference, $\mathbb{E}(|\tilde{Y}|, |\tilde{Y}_{\text{ref}}|)$ where capital $\tilde{Y}$ denotes the bandlimited spectrum;
**ESR-T**: the ESR w.r.t. the bandlimited teacher as used in the loss function (17), $\mathbb{E}(y, \tilde{y}_{\text{teach}})$;
**NMR-T**: the NMR w.r.t. the bandlimited teacher as used in the loss function (17), $\text{NMR}(S, \tilde{S}_{\text{teach}})$;
**HESR-T**: the ESR of the magnitude of the harmonic components w.r.t. the teacher, $\mathbb{E}(|\tilde{Y}|, |\tilde{Y}_{\text{teach}}|)$;
**ESR-S**: the ESR of the signal w.r.t. to the bandlimited version of itself, $\mathbb{E}(y, \tilde{y})$;
**NMR-S**: the NMR of the signal w.r.t. to the bandlimited version

of itself, $\mathrm{NMR}(S, \tilde{S})$.
In the calculations above, where the reference device was not available, the bandlimited teacher model output was used as the reference, i.e. $y_{\mathrm{ref}} := \tilde{y}_{\mathrm{teach}}$.

## 3. EXPERIMENTS

To test our methodology, we implement the teacher-student fine tuning on a total of eight neural models: two open-weight LSTMs, two open-weight TCNs and four models pre-trained by us using data from two different analog distortion effects units (available open-weight at the accompanying webpage).

### 3.1. Open-weight models

The GuitarML Tone Library[2] contains open-weight models of various distortion effects, each consisting of an LSTM unit with hidden size 40 followed by a linear layer, with a residual connection between the input and output [6]. Out of the 18 "snapshot" (non-conditioned) models on the homepage, the two "worst case" models exhibiting the most aliasing (highest NMR-S) were selected as case studies. We refer to these models as *Mesa* and *Goat*.

Neural Amp Modeller (NAM)[3] is an open-source plug-in and framework for training models with hundreds of open-weight models available on Tone3000 [4]. The two TCN models were selected as those which exhibited the most aliasing out of the ten all-time most downloaded packages (each of which contains several snapshots). The architecture of both is the NAM default: two TCN blocks each with a kernel size of 3, 10 layers, a dilation-growth of 2 and $\tanh$ activation functions. The blocks have channel widths of 16 and 8 respectively. According to the metadata these models were originally trained at $F_s = 48\,\mathrm{kHz}$, so fine-tuning was implemented at this same rate. The NAM PyTorch implementation was used with no modifications to the model code. Here we refer to our chosen TCN models as *Vox* and *JCM* with reference to the amplifiers on which they were modelled.

### 3.2. Custom models

We also trained from scratch models of two analog devices: the Hudson Broadcast germanium pre-amp pedal; and the Dunlop-MXR JHM8 Jimi Hendrix Gypsy Fuzz pedal. For each device we trained both an LSTM model and a TCN model with the architectures described in Sec 3.1. We used the training signal provided by GuitarML, consisting of chirps, noise bursts, and guitar and bass playing amounting to 3 minutes and 40 seconds of audio with $F_s = 44.1\,\mathrm{kHz}$. The LSTM training used TBPTT with a warm-up of 1000 samples and a frame-size of 2048 samples. The TCN training used a signal length of 16384 samples. In both cases the batch size was 40 and the models were trained for a maximum of 5k epochs. The loss function was a combination of ESR and DC loss, with A-weighting pre-emphasis filtering [28].

## 4. OBJECTIVE RESULTS

This section presents the objective results along with analysis of example spectra of model outputs. The metrics described in Sec. 2.6 are reported in Table 1 for all the models considered.

[2]https://guitarml.com/tonelibrary/tonelib-pro.html
[3]https://www.neuralampmodeler.com/
[4]https://www.tone3000.com/

### 4.1. Custom pre-training results

The metrics measured on our custom pre-trained models are shown in rows 1, 3, 5 and 7 of Table 1 (Broadcast/JHM8 Teacher). In terms of G+B ESR and G+B MRSL, the LSTM models gave the better result over the TCN models. Between the two devices, the JHM8 proved the easier device to model, with the JHM8 LSTM giving overall the best result (an ESR of 0.3%). The sine tone dataset metrics with respect to the reference (-R) are generally higher (worse), which is perhaps unsurprising as there were no pure sine tones in the training data (but there were sine sweeps). In all four models, the NMR-R is greater than zero, suggesting the model outputs are significantly noisier than outputs from the reference device. The NMR-R, however, will also pick up differences between non-noise components.

### 4.2. Fine-tuning results – aliasing reduction

The spectral response to a sinusoidal input can be seen in Fig. 2 for the Broadcast and JHM8 models and Fig. 3 for the open-weight models. In all cases, a reduction in aliasing can clearly be observed between the Teacher and Student models. The reduction in aliasing is especially visible in lower frequencies. Aliases below the fundamental are most likely to be audible [8], so it is reassuring that these appear to be the most suppressed. In Fig. 2a, for example, the most prominent sub-fundamental aliases have been reduced in magnitude by approximately $40\,\mathrm{dB}$. This suppression can also be seen in Fig. 2b and Fig. 3, with the most extreme example shown in Fig. 3a for the *Goat* LSTM model.

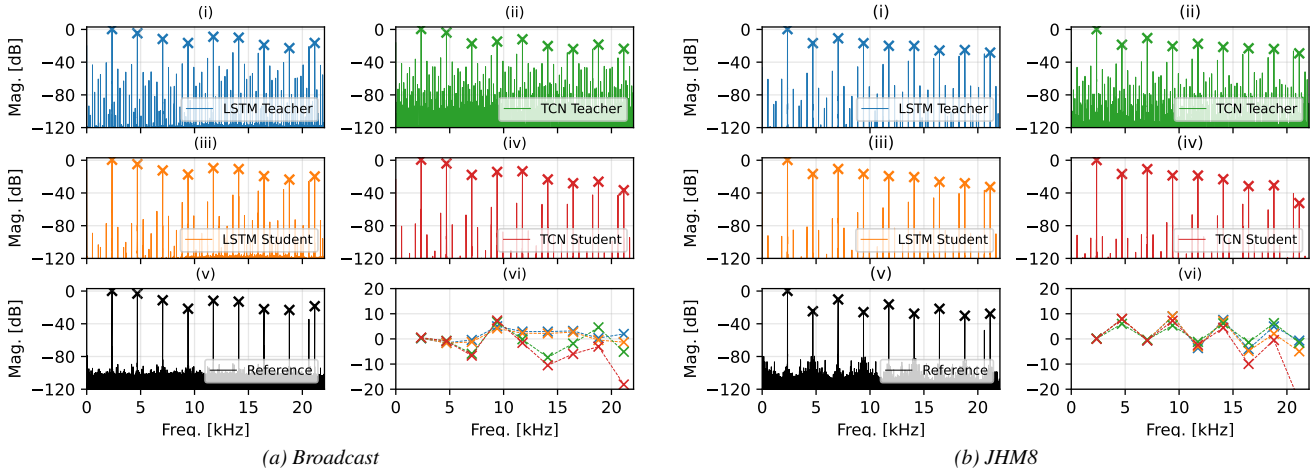
The NMR-S results in Table 1 provide a more objective (yet perceptually informed) metric of aliasing. For all eight models, the fine-tuning process results in a decrease in mean NMR-S across the sinusoids dataset. Because this is an arithmetic mean across all frequencies and input gains, it is useful to examine how NMR-S varies with input frequency – as shown in Fig. 4a-i for the Broadcast models and 4b-i for the JHM8 models. In Fig. 4a-i, for example, the NMR of the LSTM and TCN Teacher models both follow a similar trajectory; with results exceeding $-10\,\mathrm{dB}$ (an approximate threshold of aliasing audibility [8, 9]) for $f_0 \gtrapprox 1\,\mathrm{kHz}$. The proposed fine tuning method results in the NMR-R of the Student models being reduced to below $-10\,\mathrm{dB}$ for all $f_0$.

### 4.3. Fine tuning results – harmonic analysis

While it is clear that the proposed method reduces aliasing, it is important to analyse how the desired harmonic distortion components are affected by the fine-tuning process. Ideally, the process would remove all aliasing whilst retaining the exact same amplitudes of harmonics. In practice, there is inevitably some side-effect on the harmonics. The proposed loss function (17) was chosen so that it not only penalises aliasing but changes in harmonic distortion components with respect to the Teacher model. The ESR-T and NMR-T are those used in the loss function, so it is interesting to observe how these vary between the Teacher and Student models in Table 1. In all cases, the fine-tuning process results in an increase in ESR-T and a decrease in NMR-T. Ideally, these should both decrease, but it appears that the models are trading off between these two measures in training. One would hope that this means that the Student model is learning to "shift" the aliasing/noise from being perceptible (measured by the NMR) to less perceptible (measured by the ESR). However, as shown in the HESR-R results, there is always some change in the amplitudes of the harmonic components.

Table 1: *Mean signal metrics in decibels (lower better) over the audio dataset (col. 4-5) and sine tone dataset (col. 6-15). The sine tone metrics displayed are the arithmetic mean over all input $f_0$ and amplitude. Bolding indicates the best result between the teacher and student models for a given Device and Model.*

| | | | Audio dataset | | Sine tone dataset | | | | | | | | 2x oversampled | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Device | Model | Role | G+B ESR | G+B MRSL | ESR-R | NMR-R | HESR-R | ESR-T | NMR-T | HESR-T | ESR-S | NMR-S | ESR-S-OS2 | NMR-S-OS2 |
| Broadcast | LSTM | Teacher | **-15.4** | **-18.1** | -11.7 | 8.8 | -14.2 | **-25.5** | 6.1 | $-\infty$ | -25.5 | 6.1 | -44.1 | -3.4 |
| | | Student | -12.1 | -12.1 | **-12.6** | **-1.8** | **-15.4** | -19.2 | **-12.6** | -21.3 | **-34.7** | **-16.7** | -52.3 | -23.5 |
| Broadcast | TCN | Teacher | **-12.1** | **-13.3** | **-12.5** | 7.1 | **-14.6** | **-34.6** | 6.5 | $-\infty$ | -34.6 | 6.5 | -42.3 | -2.1 |
| | | Student | -7.0 | -8.3 | -8.2 | **0.5** | -9.5 | -11.8 | **-1.8** | -13.4 | **-49.3** | **-18.2** | -55.2 | -27.6 |
| JHM8 | LSTM | Teacher | **-24.1** | **-24.7** | **-17.4** | 3.3 | **-19.2** | **-42.7** | -8.9 | $-\infty$ | -42.7 | -8.9 | -57.6 | -24.8 |
| | | Student | -20.1 | -20.4 | -16.2 | **3.1** | -17.0 | -26.2 | **-16.6** | -27.8 | **-46.8** | **-22.7** | -57.9 | -28.9 |
| JHM8 | TCN | Teacher | **-19.9** | **-21.8** | **-12.0** | 3.9 | -16.1 | **-42.1** | -1.0 | $-\infty$ | -42.1 | -1.0 | -54.9 | -11.8 |
| | | Student | -13.5 | -14.1 | -11.7 | **-1.3** | **-16.7** | -22.8 | **-8.5** | -24.5 | **-52.6** | **-25.1** | -58.9 | -29.1 |
| Goat | LSTM | Teacher | $-\infty$ | $-\infty$ | $-\infty$ | **-122.0** | $-\infty$ | **-29.9** | 11.5 | $-\infty$ | -29.9 | 11.5 | -35.6 | 0.4 |
| | | Student | -11.1 | -9.3 | -11.9 | -7.7 | -12.6 | -12.0 | **-6.7** | -12.6 | **-40.8** | **-5.6** | -41.6 | -5.6 |
| Mesa | LSTM | Teacher | $-\infty$ | $-\infty$ | $-\infty$ | **-126.2** | $-\infty$ | **-29.4** | 19.3 | $-\infty$ | -29.4 | 19.3 | -46.7 | 4.2 |
| | | Student | -12.7 | -11.3 | -17.3 | -8.0 | -20.8 | -17.7 | **-10.3** | -20.8 | **-36.4** | **-14.8** | -58.2 | -25.3 |
| Vox | TCN | Teacher | $-\infty$ | $-\infty$ | $-\infty$ | **-125.4** | $-\infty$ | **-36.8** | 15.0 | $-\infty$ | -36.8 | 15.0 | -45.2 | 6.5 |
| | | Student | -8.1 | -10.4 | -17.6 | -4.1 | -20.2 | -17.8 | **-9.0** | -20.2 | **-53.5** | **-21.1** | -56.4 | -24.3 |
| JCM | TCN | Teacher | $-\infty$ | $-\infty$ | $-\infty$ | **-124.3** | $-\infty$ | **-33.6** | 13.4 | $-\infty$ | -33.6 | 13.4 | -46.1 | 3.0 |
| | | Student | -10.0 | -13.2 | -18.1 | -5.2 | -22.6 | -18.2 | **-9.4** | -22.6 | **-44.9** | **-12.5** | -58.7 | -24.6 |




*(a) Broadcast* *(b) JHM8*

Figure 2: *Output magnitude spectra of the Broadcast (a) and JHM8 (b) Teacher models (i-ii), their respective Student models (iii-iv) and the reference (v) for an input tone of* 2394.3 Hz. *Crosses mark the harmonic components, and (vi) shows the error in magnitude of these w.r.t. the reference.*

The effect on the harmonic components for a sinusoidal input can be seen in Figures 2 and 3. The general trend – especially noticeable in Fig. 3 – is that the anti-aliasing procedure results in a damping of high frequency harmonics. For frequency components above 12 kHz, this is to be fully expected due to the pre-emphasis LPF used during training – errors in high frequency harmonics are ignored by design. Errors in harmonics within the more sensitive human hearing bands are more critical as they may be perceived as a difference in the desired harmonic distortion.

For the Broadcast and JHM8 models, the magnitude error in harmonic components with respect to the reference (the HESR-R) is shown in Table 1. Interestingly, in some cases (Broadcast LSTM and JHM8 TCN) the Student models achieve a lower (better) HESR-R than their respective Teacher models, which is a remarkable result considering the Student models were shown no additional data from the reference device during fine-tuning. In the other two cases, however, the result is the opposite and the largest discrepancy can be seen between the Broadcast TCN models. Fig. 4a-ii shows HESR-R against input sinusoidal frequency for the Broadcast models. The results may appear similar at first glance, but there is a large discrepancy between the TCN Teacher and Student models for $f_0$ around 200 Hz – a perceptually important frequency range for guitar and bass processing.

### 4.4. Comparison with oversampling

It is interesting to compare the results of the fine-tuning process with that of oversampling the original pre-trained model. Here we use an oversampling factor of two, and implement this for both the Teacher and Student models in all cases. The oversampled LSTMs were implemented via the method in [21]. For the TCN models, oversampling was implemented by upsampling the convolutional kernels, i.e. increasing the convolution dilation of each layer by a factor of 2. In all cases, frequency-domain resampling was used to convert the sample rate of the input/output signals to/from the oversampled rate (see e.g. [34]).

The ESR-S and NMR-S for the oversampled models are

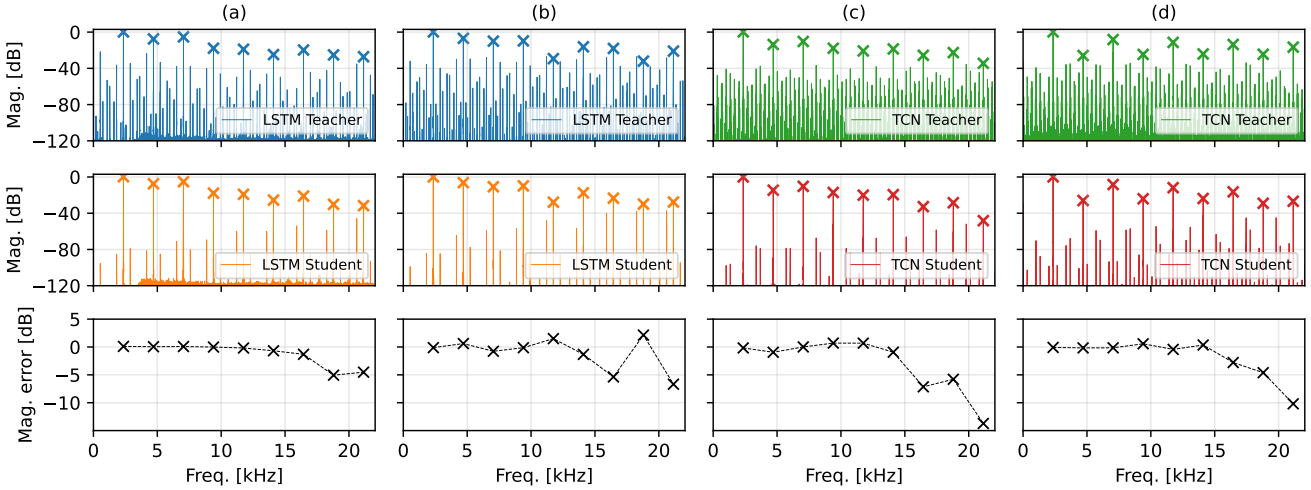
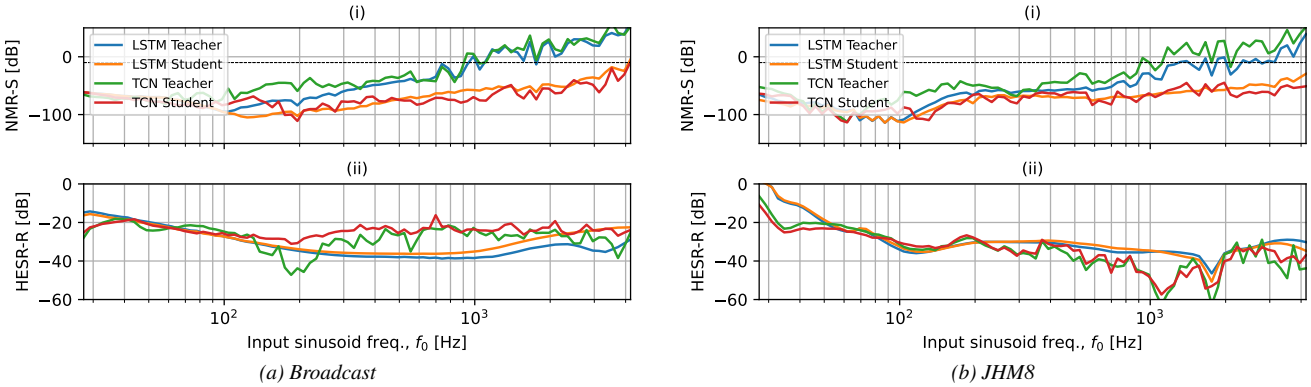



Figure 3: *Output magnitude spectra of the Teacher models (top), the corresponding Student models (middle) and the relative error in magnitude of the harmonic components (bottom) for the open-weight Goat (a), Mesa (b), Vox (c) and JCM (d) models. The input tone had* $f_0 = 2394.3\,\text{Hz}$ *and amplitude* $-6\,\text{dB}$.




*(a) Broadcast* *(b) JHM8*

Figure 4: *NMR-S (i) and HESR-R (ii) against input sinusoidal frequency,* $f_0$*, for the Broadcast (a) and JHM8 (b) models. The input gain was* $-6\,\text{dB}$. *The black dashed line at* $-10\,\text{dB}$ *indicates the approximate threshold of aliasing audibility [8]. Lower values are better.*

shown in the last two columns of Table 1. In all but one case (JHM8 LSTM) the proposed fine-tuned model performs better in terms of aliasing reduction than 2x oversampling of the original model. The fine-tuning process requires extra resources during training, but at inference it requires no extra operations per sample compared to the original model, unlike oversampling. Fig. 5 shows a sine sweep passed through the Broadcast LSTM Teacher and Student models both at the base and oversampled rates.

## 5. PERCEPTUAL EVALUATION

A Multiple Stimuli Hidden Reference and Anchor (MUSHRA) [35] listening test was conducted to investigate how the original pre-trained models (Teacher) and the fine-tuned (Student) models were perceived compared to the reference analog audio device. Only the *Broadcast* and *JHM8* models were included in the test; the others were excluded due to the lack of reference audio data. The anchor was the input signal hard-clipped between -0.5 and 1.0 with an input and output gain of $48\,\text{dB}$ and $-9\,\text{dB}$ respectively.

Six input signals were used to generate the test excerpts: two direct-input (DI) electric guitar clips, two DI bass clips, a linear sine sweep from 20 Hz to 10 kHz and a linear sine sweep from 10 kHz to 20 kHz. Across the two reference devices, this gave 12 trials per test. For each trial, listeners were presented with an audio clip from the reference device and asked to rate the four model outputs (LSTM Teacher, LSTM Student, TCN Teacher and TCN Student), the Hidden Reference and the Anchor based on their perceived similarity to the reference.

Sixteen volunteers participated in the test, all of which were either students, academics, professionals or practitioners within audio technology. Participants who rated the Reference < 80% in > 15% of trials were excluded from the analysis, leaving 12 remaining. This is a relaxation of the MUSHRA standard post-screening guidance [35] in which a similarity threshold of 90% is recommended, but using this criterion there would have only been six participants remaining.

Violin plots of the MUSHRA test results aggregated across participants and both devices are shown in Fig. 6. The medians with 95% confidence intervals (CI) are displayed.

While the participants were generally capable of identifying

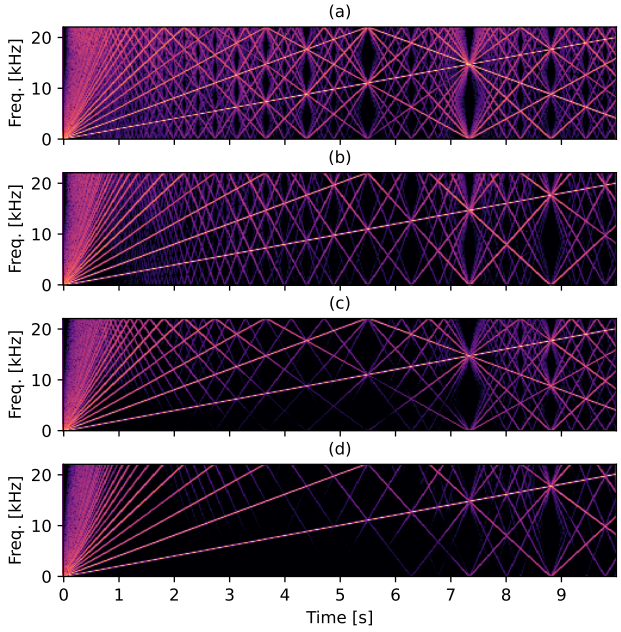



Figure 5: *Response to a sine sweep for the Broadcast LSTM models: (a) Teacher (b) Teacher 2x oversampled (c) Student (d) Student 2x oversampled. The minimum amplitude visible is* $-80\,\mathrm{dB}$.

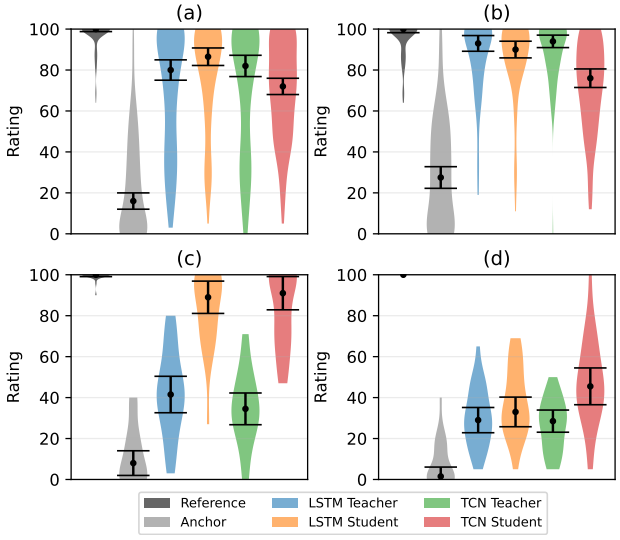



Figure 6: *Perceived similarity ratings across (a) all samples combined, (b) guitar and bass samples, (c) the sine sweep from 0 to 10kHz and (d) the sine sweep from 10k-20kHz. Results from the Broadcast and JHM8 pedals are aggregated together. Black dots and error bars show the median and 95% confidence interval.*

the Reference, there were some cases where they rated it less than 100 – suggesting confusion over which was the correct Reference. These cases were much more common for guitar and bass inputs than for the sweeps. The Anchor results show a large spread, and again this was input dependent. For the sweep inputs, the Anchor was exclusively rated lower than 40 with the median and 95% confidence interval below 20 ("Very Poor" perceived similarity). For guitar and bass inputs, the results were higher and in some cases the anchor was rated very highly – suggesting perhaps this was not the best choice of anchor for the experiment.

The results for the original Teacher models (both LSTM and TCN) show a large spread of results, and a bimodal distribution is observed in Fig. 6a. For guitar and bass inputs, the median CIs for both the LSTM Teacher and TCN model lie above 85% – indicating that both models give an Excellent perceived similarity to the reference. Since the reference contains no aliasing, it appears that aliasing caused by the original models was not noticeable for the guitar and bass inputs. When driven with the sine sweeps however, the perceived similarity is significantly lower with the median CI within the Very Poor/Fair rating bands. Participants reported that under these conditions, it was much easier to distinguish models from the reference due to the presence of aliasing artefacts.

The results for the fine-tuned (Student) models also show a distinction between the input stimuli. For the sine sweep up to 10k, there is a significant increase in perceived similarity compared to the Teacher models, with the median and CI in the Excellent range for both LSTM and TCN Student models. This shows that the proposed models have been effective at reducing perceived aliasing for inputs within this bandwidth. For frequencies above $10\,\mathrm{kHz}$, however, there is less improvement. The TCN Student model shows a statistically significant improvement over its Teacher model but the median CI is only within the Fair band, whereas for the LSTM there is very little improvement. Considering the models were trained on sinusoids up to a maximum $f_0$ of $12\,\mathrm{kHz}$, it is not surprising that aliasing is still noticeable for input frequencies beyond this. On the guitar and bass stimuli, the LSTM Student model shows no significant change in perceived quality compared to its Teacher. This is a good result: where the original model performed well, its performance has not been affected by fine-tuning; but where aliasing was originally a problem, the fine-tuning process has reduced aliasing (at least for inputs up to $10\,\mathrm{kHz}$). The TCN Student model results, however, show a significant reduction in perceived similarity to the Reference for guitar and bass inputs. It was found that the measured suppression in high frequencies (Sec. 4) were indeed perceptible, as some participants who reported that the TCN Student models sounded "low-passed" or "less distorted" than the reference. However, the median TCN Student score and CI was still within the Very Good range.

## 6. CONCLUSIONS AND FURTHER WORK

This work presented a fine-tuning procedure for reducing the aliasing caused by neural network models of guitar distortion effects. This involved training a Student model against an aliasing-free synthetic dataset –generated during training by processing sinusoids through the original pre-trained (Teacher) model and then removing non-harmonic components through Fourier analysis and re-synthesis. As case studies, open-weight LSTM and TCN models were considered, and an example of each trained from scratch on two analog fuzz effects pedals. The proposed method consistently reduced aliasing across all systems, outperforming two times oversampling in all but one case. However, fine-tuning sometimes altered desirable harmonic content. A MUSHRA listening test was deployed to evaluate how the original pre-trained (Teacher) models and the fine-tuned (Student) models compared in perceived similarity to two analog reference devices. It was found that for sine sweep inputs – for which lots of aliasing was

present in the Teacher outputs – fine-tuning significantly improved the similarity score for both LSTM and TCN models. For non-sinusoidal guitar and bass signals, there was no significant difference between the LSTM Student model and its Teacher, with both rated as Excellent in similarity to the reference. For the TCN models, there was a reduction in perceived similarity from Excellent to Very Good, indicating that fine-tuning had an adverse effect on the desired harmonic distortion. While our results show the potential of the proposed method for anti-aliasing in neural networks, there are still areas for further work, particularly regarding the affected harmonic distortion components. For example a hyper-parameter sweep of loss function weighting $\lambda$, a comparison of different pre-emphasis filters or an investigation into corrective filters post-training.

## 7. REFERENCES


[1] V. Välimäki, S. Bilbao, J. O Smith, J. S Abel, J. Pakarinen, and D. Berners, "Virtual analog effects," in *DAFX: Digital Audio Effects*, U. Zölzer, Ed., pp. 473–522. John Wiley & Sons, Ltd, 2011.

[2] M. Karjalainen and J. Pakarinen, "Wave digital simulation of a vacuum-tube amplifier," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Toulouse, France, May 2006, pp. 153–156.

[3] R. C. D. Paiva, S. D'Angelo, J. Pakarinen, and V. Välimäki, "Emulation of operational amplifiers and diodes in audio distortion circuits," *IEEE Trans. Circ. Syst. II*, vol. 59, no. 10, pp. 688–692, Oct. 2012.

[4] F. Eichas and U. Zölzer, "Black-box modeling of distortion circuits with block-oriented models," in *Proc. 19th Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, Sept. 2016, pp. 39–45.

[5] A. Wright, V. Välimäki, and E.-P. Damskägg, "Real-time black-box modelling with recurrent neural networks," in *Proc. 22nd Int. Conf. Digital Audio Effects*, Birmingham, UK, Sept. 2019.

[6] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time guitar amplifier emulation with deep learning," *Appl. Sci.*, vol. 10, no. 2, 2020.

[7] L. Juvela, E.-P. Damskägg, A. Peussa, Jaakko Mäkinen, T. Sherson, S. Mimilakis, K. Rauhanen, and A. Gotsopoulos, "End-to-end amp modeling: from data to controllable guitar amplifier models," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Rhodes, Greece, 2023.

[8] H.-M. Lehtonen, J. Pekonen, and V. Välimäki, "Audibility of aliasing distortion in sawtooth signals and its implications for oscillator algorithm design," *J. Acoust. Soc. Am.*, vol. 132, no. 4, pp. 2721–2733, Oct. 2012.

[9] J. Kahles, F. Esqueda, and V. Välimäki, "Oversampling for nonlinear waveshaping: Choosing the right filters," *J. Audio Eng. Soc.*, vol. 67, no. 6, pp. 440–449, Jun. 2019.

[10] F. Esqueda, V. Välimäki, and S. Bilbao, "Aliasing reduction in soft-clipping algorithms," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 2014–2018.

[11] J. D Parker, V. Zavalishin, and E. Le Bivic, "Reducing the aliasing of nonlinear waveshaping using continuous-time convolution," in *Proc. 19th Int. Conf. on Digital Audio Effects (DAFx-16)*, 9 2016.

[12] S. Bilbao, F. Esqueda, J. D. Parker, and V. Valimaki, "Antiderivative antialiasing for memoryless nonlinearities," *IEEE Signal Processing Letters*, vol. 24, pp. 1049–1053, 2017.

[13] S. Bilbao, F. Esqueda, and V. Välimäki, "Antiderivative antialiasing, Lagrange interpolation and spectral flatness," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 10 2017, pp. 141–145.

[14] V. Zheleznov and S. Bilbao, "Interpolation filters for antiderivative antialiasing," in *Proc. 27th Int. Conf. Digital Audio Effects (DAFx24)*, Guildford, UK, Sept. 2024.

[15] M. Holters, "Antiderivative antialiasing for stateful systems," in *Proc. 22nd Int. Conf. on Digital Audio Effects (DAFx-19)*, sept 2019.

[16] M. Holters, "Antiderivative antialiasing for stateful systems," *Applied Sciences*, vol. 10, no. 1, 2020.

[17] D. Albertini, A. Bernardini, and A. Sarti, "Antiderivative antialiasing in nonlinear wave digital filters," in *Proc. 23rd Int. Conf. on Digital Audio Effects (DAFx-20)*, 9 2020.

[18] T. Vanhatalo, P. Legrand, M. Desainte-Catherine, P. Hanna, and G. Pille, "Evaluation of real-time aliasing reduction methods in neural networks for nonlinear audio effects modelling," *Journal of the Audio Engineering Society*, vol. 72, pp. 114–122, 3 2024.

[19] L. Köper and M. Holters, "Antialiased state trajectory neural networks for virtual analog modeling," in *Proc. 26th Int. Conf. on Digital Audio Effects (DAFx23), Copenhagen, Denmark, September 2023*, 2023.

[20] J. Chowdhury, "Sample-rate agnostic recurrent neural networks," https://jatinchowdhury18.medium.com/sample-rate-agnostic-recurrent-neural-networks-238731446b2, Apr. 2022. Accessed 5/3/24.

[21] A. Carson, A. Wright, J. Chowdhury, V. Välimäki, and S. Bilbao, "Sample rate independent recurrent neural networks for audio effects processing," in *Proc. 27th Int. Conf. Digital Audio Effects (DAFx24)*, Guildford, UK, Sept. 2024.

[22] A. Carson, V. Välimäki, A. Wright, and S. Bilbao, "Resampling filter design for multirate neural audio effect processing," *arXiv pre-print 2501.18470*, Jan. 2025.

[23] E.-P. Damskägg, L. Juvela, E. Thuillier, and V. Välimäki, "Deep learning for tube amplifier emulation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP19)*, Brighton, UK, May 2019, pp. 471–475.

[24] E.-P. Damskägg, L. Juvela, and Vesa Välimäki, "Real-time modeling of audio distortion circuits with deep learning," in *Sound and music computing conference*, 2019, pp. 332–339.

[25] A. Wright, A. Carson, and L. Juvela, "Open-amp: Synthetic data framework for audio effect foundation models," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Hyderabad, India, Apr. 2025.

[26] F.J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[27] V. Zheleznov, "Interpolation filters for antiderivative antialiasing," M.S. thesis, University of Edinburgh, 2023.

[28] A. Wright and V. Välimäki, "Perceptual loss function for neural modeling of audio systems," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2020, pp. 251–255.

[29] *Method for objective measurements of perceived audio quality*, ITU-R recommendation BS.1387, 1998.

[30] P. Kabal, "An examination and interpretation of ITU-R BS.1387: Perceptual evaluation of audio quality," Tech. report, McGill Univ., 2003.

[31] M. Moritz, Toni Olán, and Tuomas Virtanen, "Noise-to-mask ratio loss for deep neural network based audio watermarking," in *IEEE 5th International Symposium on the Internet of Sounds, IS2 2024*, 8 2024.

[32] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Barcelona, Spain, 2020, pp. 6199–6203.

[33] C. Steinmetz and J. D. Reiss, "auraloss: Audio focused loss functions in PyTorch," in *Digital Music Research Network One-day Workshop (DMRN+15)*, 2020.

[34] V. Välimäki and S. Bilbao, "Giant FFTs for sample-rate conversion," *J. Audio Eng. Soc.*, vol. 71, pp. 88–99, Mar. 2023.

[35] *Method for the subjective assessment of intermediate quality level of audio systems*, ITU-R recommendation BS.1534-3.