ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10097173

# MODELLING BLACK-BOX AUDIO EFFECTS WITH TIME-VARYING FEATURE MODULATION

*Marco Comunità*[1] *Christian J. Steinmetz*[1] *Huy Phan*[2*] *Joshua D. Reiss*[1]

[1]Centre for Digital Music, Queen Mary University of London, UK
[2]Amazon Alexa, Cambridge, MA, USA

## ABSTRACT

Deep learning approaches for black-box modelling of audio effects have shown promise, however, the majority of existing work focuses on nonlinear effects with behaviour on relatively short time-scales, such as guitar amplifiers and distortion. While recurrent and convolutional architectures can theoretically be extended to capture behaviour at longer time scales, we show that simply scaling the width, depth, or dilation factor of existing architectures does not result in satisfactory performance when modelling audio effects such as fuzz and dynamic range compression. To address this, we propose the integration of time-varying feature-wise linear modulation into existing temporal convolutional backbones, an approach that enables learnable adaptation of the intermediate activations. We demonstrate that our approach more accurately captures long-range dependencies for a range of fuzz and compressor implementations across both time and frequency domain metrics. We provide sound examples, source code, and pretrained models to faciliate reproducibility[1].

***Index Terms***— Audio effects, black-box modelling, modulation

## 1. INTRODUCTION

Audio effects are tools employed by audio engineers and musicians central to shaping the timbre, dynamics, and spatialisation of sound [1]. Digital emulation of audio effects, often referred to as virtual analogue, is an area of active research [2–6] with methods often categorised into white-, grey- and black-box approaches. White-box modelling relies on complete knowledge of the system and often employs differential equations, which enables high quality emulations but often entails a time consuming design process and computationally expensive models [7, 8]. Grey-box approaches [9, 10] combine a partially theoretical model with input-output measurements. This greatly reduces the prior knowledge necessary to model a device while maintaining interpretability, but still requires understanding of the underlying implementation and carefully designed measurement and optimisation procedures. This motivates black-box models that enable emulations using only measurements from the device. Recently, deep learning approaches have seen success in modelling a range of effects [11–14]. These approaches often leverage recurrent or convolutional networks operating in the time domain [15–18]. While successful for some effects, modelling behaviours on longer time scales has proven challenging and is so far less investigated.

In this work, we focus on nonlinear time-varying audio effects that exhibit input-dependant behaviour over long time scales, such as fuzz distortion and dynamic range compression. Distortion effects often present a challenge due to their highly nonlinear behaviour, which has been addressed in previous works relying on convolutional networks with short receptive field [16] or simple one-layer recurrent networks [18]. However, distortion effects such as fuzz can also pose an additional challenge since they exhibit time-varying behaviour over larger time scales due to the attack and release of the circuit. Fuzz is characterised not only by asymmetrical clipping, which for sinusoidal inputs results in a rectangular wave output, but also for its attack and release time constants which modulate the behaviour of the device as a function of the input. This results in a characteristic time-varying distortion, which existing deep learning based approaches fail to accurately capture, as shown in Fig. 1.

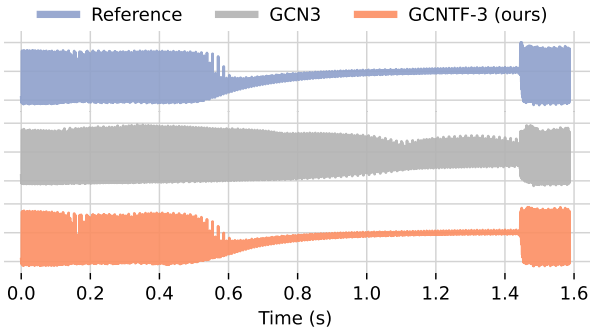


**Fig. 1**: State-of-the-art black-box models like GCN-3 [19] (grey) fail to capture the behaviour of effects with large time constants such as fuzz (blue). Our proposed approach GCNTF-3 (orange), which extends previous convolutional networks with time-varying feature modulation, enables accurate modelling of this behaviour.

Dynamic range compressors also exhibit time-varying nonlinear behaviour over a range of timescales. In some cases the release times of compressors can reach several seconds, such as in the classic LA-2A compressor. Modelling of compression has been addressed employing a range of strategies including convolutional-recurrent architectures [20], time-frequency representations through autoencoding [21], and shallow temporal convolutional networks with large receptive field [14]. However, performance when modelling configurations with large release time constants has not been investigated.

We propose a method to model the nonlinear behaviour over large time-scales such as fuzz and dynamic range compression by incorporating time-varying feature modulation (Temporal FiLM) [22] into existing temporal convolutional backbones. This enables adaptation of the activations of the network as a function of the input signal. While this is achieved through a simple mechanism of scaling and shifting the activations, we demonstrate that this enables superior performance across a range of effects without increasing the receptive field of the main network. Our contributions include the integration of temporal feature modulation into the black-box audio effect modelling framework and a set of benchmark datasets comprised of fuzz and compressor effects with varying time constants, which we utilise to demonstrate the failure modes of existing approaches and the ability of our proposed approach to address these limitations.

*Work done when H. Phan was at Centre for Digital Music, prior to joining Amazon.
[1]https://mcomunita.github.io/gcn-tfilm_page

## 2. METHOD

Audio effects are signal processing devices that given an input $x \in \mathbb{R}^L$ with $L$ samples and a set of $P$ parameters $\phi \in \mathbb{R}^P$ that control the operation of the system, output a modified version $y \in \mathbb{R}^L$ of the signal. In this work, we focus on modelling the input-output function $y = f(x, \phi)$, at one configuration of the device, holding $\phi$ constant. We aim to design a neural network $g_\theta(x)$ that produces a signal $\hat{y}$ perceptually indistinguishable from the real output $y$. The modelling process involves training $g_\theta(x)$ with a dataset of $E$ examples $\mathcal{D} = \{(x_i, y_i, \phi)\}_{i=1}^{E}$ containing input-output recordings $(x_i, y_i)$ at fixed parameters $\phi$. A loss function $\mathcal{L}(\hat{y}, y)$ measures the difference between the output of the network and the target system, providing a means to update the weights $\theta$ through a given number of gradient-based optimisation steps.

### 2.1. Modelling Network

Similar to previous work on distortion effect modelling [12, 16], we adopt a feedforward WaveNet [23] architecture, also known as a temporal convolutional network (TCN). We refer to this architecture as the Gated Convolution Network (GCN) since it a special case of the TCN that utilises gated convolutions. The GCN is composed of $M$ blocks with each block containing $\frac{N}{M}$ layers for a total of $N$ layers. Each layer in a block is made of a dilated 1-dimensional convolutional layer followed by a gated activation as shown in Fig. 2 (left). The outputs from each layer are summed through a $1 \times 1$ convolution to produce the final output $y$. We implement several variants of the base GCN architecture with short, medium and long receptive fields relying on different number of layers and kernel sizes, and make also use of rapidly growing dilation factors [14, 24, 25].

### 2.2. Temporal FiLM

Feature-wise Linear Modulation (FiLM) is a general-purpose conditioning method that operates on the intermediate features of a neural network as a function of conditioning signals [26]. Given a conditioning signal $\mathbf{x}_i$, FiLM learns two functions $f$ and $g$, which are used to map the conditioning signal to a set of scaling $\gamma_{n,c} = f(\mathbf{x}_i)$ and bias $\beta_{n,c} = h(\mathbf{x}_i)$ parameters for each layer $n$ and channel $c$ of the network. These parameters are used to modulate the intermediate activations at each layer $\mathbf{z}_{n,c}$ via a feature-wise affine transformation

$$\text{FiLM}(\mathbf{z}_{n,c}, \gamma_{n,c}, \beta_{n,c}) = \gamma_{n,c} \cdot \mathbf{z}_{n,c} + \beta_{n,c}. \quad (1)$$

In practice, $f$ and $h$ are implemented as a neural network and can be learned during training of the main network.

While FiLM has proven to be a powerful conditioning method, it can be further extended to increase the expressivity of the network by leveraging long-range dependencies in the conditioning signal to vary the modulation of intermediate features across time; an approach known as Temporal Feature-wise Linear Modulation (TFiLM) [22]. Using recurrent networks, TFiLM layers modulate the intermediate features of a convolutional model over time as a function of the activations at each layer. This has conceptual connections to other input dependant and time-varying conditioning approaches such as hypernetworks [27] and dynamic convolution [28], which enable adaptation of the weights of convolutional networks. However, TFiLM provides a simpler method for adaptation that is both efficient and often easier to train. Thus far, TFiLM has only been applied to the task of audio super resolution using UNet-like architectures and has not yet been integrated in the GCN/TCN architecture, as we propose in this work.

To capture both nonlinear behaviour and long range temporal dependencies in modelling audio effects, we propose to integrate time-varying feature-wise linear modulation in the base GCN architecture, which we refer to as GCN with Temporal FiLM (GCNTF). In our formulation, given a sequence of activations $\mathbf{z}_n \in \mathbb{R}^{C \times L}$ from the $n$-th layer of a GCN, where $C$ is the number of channels, and $L$ is the sequence dimension, we split the sequence into $T$ blocks of $B$ samples along the sequence dimension. For each block $\mathbf{z}_n^t$, 1-dimensional max pooling is applied to downsample the signal by a factor of $B$ to produce $\mathbf{a}_n^t$ as shown in Fig. 3. Then an LSTM generates a sequence of scaling and bias parameters $\text{LSTM}_n(\mathbf{a}_n^t) = (\gamma_{n,0}^t, ..., \gamma_{n,c}^t), (\beta_{n,0}^t, , ..., \beta_{n,c}^t)$ for each channel $c$. These scaling and bias parameters are then used to modulate each channel of the activations individually in each block by an affine transformation

$$\mathbf{y}_{n,c}^t = \gamma_{n,c}^t \mathbf{z}_{n,c}^t + \beta_{n,c}^t. \quad (2)$$

As shown in Fig. 2 (right), the output of each TFiLM module is sent through a $1 \times 1$ convolution and combined with the residual connection $\mathbf{x}_n$ and sent to the following layer. The output of the TFiLM module $y_n$ is sent to the final layer of the network where all intermediate outputs are mixed together via another $1 \times 1$ convolution.

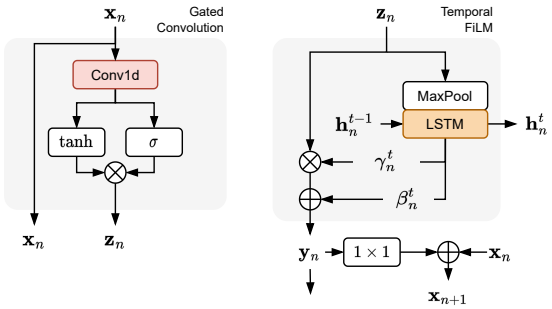


**Fig. 2**: Block diagram of the dilated 1-dimensional gated convolution block (left) and the Temporal FiLM module (right).

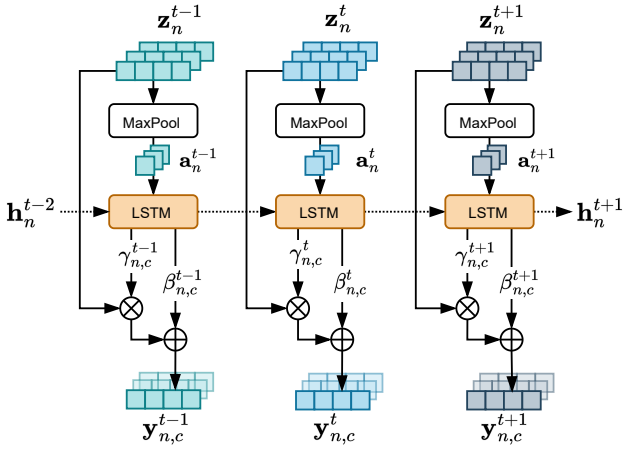


**Fig. 3**: Temporal FiLM modulates the intermediate activations of a convolutional network at each layer by splitting feature maps along the sequence dimension into $T$ blocks $\mathbf{z}_n^t$. Max pooling is applied to each block to generate $\mathbf{a}_n^t$, which is used as input to the LSTM (with hidden activations $\mathbf{h}_n^t$) that generates scaling and bias parameters. This illustrates a case when $B = 4$ and $C = 3$. For clarity only the affine transformation of the first channel is shown.

## 3. EXPERIMENTAL DESIGN

To understand how Temporal FiLM aids in the modelling process, specifically for effects with behaviour at long time scales, we train models across two effect classes: fuzz and compressor. As baselines, we considered state-of-the-art convolutional networks [15, 16, 19] proposed for guitar amplifier and distortion effect modelling. The models (GCN-1 and GCN-3) have respectively, 1 block of 10 layers and 2 blocks of 9 layers, dilation growth of 2, kernel size of 3 and 16 channels for every convolutional layer; giving receptive fields of 2047 and 2045 samples ($\approx 45$ ms at $f_s = 44.1$ kHz).

While GCN-1 and GCN-3 were successful in modelling guitar amplifiers and distortion effects, they may not be capable of modelling effects with longer temporal behaviour due to their relatively small receptive field. To address this, we construct stronger baselines by extending these models to create variants with longer receptive fields by adopting larger dilation growth [14]. As a result, GCN-250 has 1 block of 4 layers, kernel size of 41, and dilation growth of 6, for a receptive field of 250 ms, while GCN-2500 has 1 block of 10 layers, kernel size of 5 and dilation growth of 3 for a receptive field of 2500 ms. As further baselines, we also include state-of-the-art recurrent networks (LSTM-32 and LSTM-96) [19]. To validate our approach we then added TFiLM layers to each of the baseline models to enable time-varying feature modulation. We refer to these models as: GCNTF-1, GCNTF-3, GCNTF-250, GCNTF-2500, which results in a total of 12 different models that we considered.

### 3.1. Experiments

**Time constants —** To evaluate the ability of the models to capture behaviour over long time scales, we created a set of specialised datasets for fuzz and compressor effects. For fuzz, we designed an analogue circuit (Custom Fuzz) which includes, together with the typical volume and gain, attack and release controls, which was designed using LiveSpice[2]. For the compressor, we used the implementation in the Pedalboard library[3], which enables arbitrary control over the time constants. With these implementations, we then assembled a dataset of processed electric guitar signals using recordings from a subset of the IDMT-SMT-Guitar dataset [29], which contains short musical pieces recorded with two different guitars, for a total of $\approx$ 28 min of audio. Clean and processed audio were split in $\approx$ 14 min training data and $\approx$ 7 min each for validation and test. To make sure to capture the complex dynamic behaviour of our design the input signal's amplitude changes randomly every 5 sec. For Fuzz, each model was trained with 3 different settings for attack and release times, respectively: 50 ms and 50 ms, 10 ms and 250 ms, 1 ms and 2500 ms, while for compressor attack and release were set to, respectively: 10 ms and 50 ms, 5 ms and 250 ms, 1 ms and 2500 ms.

**Other effects —** To further demonstrate the performance of our approach we also trained models on the Fuzz Face emulation plugin from Distorque Audio[4], the LA-2A compressor from the SignalTrain dataset [21], and digital compressor, MCompressor, by Melda Production[5]. For the MCompressor, we selected two attack and release settings: 5 ms and 250 ms, 1 ms and 1000 ms. The LA-2A has no attack and release controls, but it is a complex analogue compressor design with a "*ratio of 3:1, a frequency dependent average attack time of 10 ms and a release time of about 60 ms for 50% of the release, and anywhere from 1 to 15 seconds for the rest*"[6].

[2] https://www.livespice.org
[3] https://github.com/spotify/pedalboard
[4] http://distorqueaudio.com/plugins/face-bender.html
[5] https://www.meldaproduction.com/MCompressor
[6] https://www.uaudio.com/blog/la-2a-collection-tips-tricks/


**Fig. 4**: MR-STFT error for models of fuzz (Custom Fuzz) and compressor (Pedalboard) effects with varying time constants. The best performing models for each effect configurations are in boldface.

**Channel width —** To ensure performance of models with TFiLM is not simply due to the increase in number of trainable parameters, we compared GCNTF-3 with a larger variant of the GCN-3 model with $C = 24$ channels. This results in both models having $\sim$ 71k parameters. These models were then trained on the Custom Fuzz with 1 ms attack and 2500 ms release.

**Block size —** We conclude our experiments measuring the performance of model with TFiLM as a function of the block size $B$, which relates to the downsampling factor and adaptation rate. We ran experiments with block sizes of $B \in 32, 64, 128, 256, 512$, training GCNTF-3 on the Custom Fuzz with 1 ms attack and 2500 ms release.

### 3.2. Training details

All models were trained with Adam with weight decay of $1 \cdot 10^{-4}$ and an initial learning rate of $5 \cdot 10^{-3}$. The learning rate was halved whenever the validation loss saw no improvement for 10 epochs. We used early stopping with a patience of 40 epochs on the validation loss and limited training to 2000 epochs, with most models training for less than 400 epochs and none reaching the limit. All models were trained at $f_s = 44.1$ kHz with inputs of 112640 samples ($\approx 2.5$ s) and batch size of 6. We used a combination of the error in the time and frequency domains, respectively: mean absolute error (MAE) and multi-resolution short-time Fourier Transform error (MR-STFT) [30, 31], as in previous work [14, 32]. The overall loss is a sum of two terms $\mathcal{L} = \mathcal{L}_{\text{MAE}} + \alpha \mathcal{L}_{\text{MR-STFT}}$, with $\alpha = 1$.

## 4. RESULTS

**Time constants —** Results comparing our proposed approach with the state-of-the-art for the tasks of modelling Custom Fuzz and Pedalboard compressor across attack and release settings are shown in Fig. 4. Results on fuzz are consistent, with GCNTF-3 performing best regardless of the time constants and error approximately halved with respect to the GCN-3. This result demonstrates how the very short receptive field of the two models (2045 samples) captures the rich timbre of the fuzz on a short-range, while TFiLM enables modelling long-range dependencies with the past input. Furthermore, even models with a receptive field sufficient to capture the past con-

| Model | Params. | LA-2A | | MComp. (5ms/250ms) | | MComp. (1ms/1000ms) | | Face Bender | |
|---|---|---|---|---|---|---|---|---|---|
| | | $L1$ | MR-STFT | $L1$ | MR-STFT | $L1$ | MR-STFT | $L1$ | MR-STFT |
| LSTM-32 | 4.5k | 0.012 | 0.356 | 0.001 | 0.239 | 0.002 | 0.250 | 0.004 | 0.236 |
| LSTM-96 | 38.1k | 0.012 | 0.323 | 0.002 | 0.275 | 0.002 | 0.278 | 0.026 | 0.379 |
| GCN-1 | 17.1k | 0.012 | 0.333 | 0.001 | 0.201 | 0.002 | 0.246 | 0.004 | 0.204 |
| GCN-3 | 32.0k | 0.001 | 0.331 | 0.022 | 0.197 | 0.002 | 0.255 | 0.002 | 0.192 |
| GCN-250 | 65.6k | 0.002 | 0.339 | 0.022 | 0.218 | 0.033 | 0.206 | 0.004 | 0.222 |
| GCN-2500 | 26.4k | 0.001 | 0.310 | 0.022 | 0.186 | 0.033 | 0.184 | 0.226 | 0.239 |
| GCNTF-1 (ours) | 38.9k | 0.012 | 0.306 | 0.022 | 0.195 | **0.001** | **0.176** | 0.004 | 0.224 |
| GCNTF-3 (ours) | 71.1k | 0.001 | 0.302 | 0.022 | 0.182 | 0.001 | 0.191 | **0.001** | **0.164** |
| GCNTF-250 (ours) | 74.3k | 0.011 | 0.346 | **3.0e-4** | **0.174** | 0.033 | 0.183 | 0.003 | 0.192 |
| GCNTF-2500 (ours) | 48.2k | **0.001** | **0.296** | 3.1e-4 | 0.179 | 0.033 | 0.167 | 0.225 | 0.213 |

**Table 1**: Impact of using TFiLM to model LA-2A, MCompressor and Face Bender. Lowest total error for each configuration in boldface.

| Model | $C$ | Params. | $L1$ | MR-STFT |
|---|---|---|---|---|
| GCN-3 | 16 | 31.97k | 0.045 | 0.502 |
| GCN-3 | 24 | 70.99k | 0.046 | 0.505 |
| GCNTF-3 (ours) | 16 | 71.14k | **0.011** | **0.279** |

**Table 2**: Impact of channel width $C$ in modelling the Custom Fuzz with attack of 1ms and release of 2500 ms.

| Model | $B$ | $L1$ | MR-STFT |
|---|---|---|---|
| GCN-3 | - | 0.045 | 0.502 |
| GCNTF-3 (ours) | 32 | 0.042 | 0.538 |
| GCNTF-3 (ours) | 64 | 0.013 | 0.307 |
| GCNTF-3 (ours) | 128 | **0.011** | **0.279** |
| GCNTF-3 (ours) | 256 | 0.013 | 0.288 |
| GCNTF-3 (ours) | 512 | 0.015 | 0.314 |

**Table 3**: Impact of feature modulation block-size $B$ in modelling the Custom Fuzz with attack of 1 ms and release of 2500 ms.

text do not capture the device behaviour, fully motivating models with TFiLM. To further compare the two models we propose to compute the error across time (e.g., every 8192 samples) and compare the distributions, as we do in Fig. 5 for fuzz with 2500ms release. We observe the error for GCN-3 has a higher median and heavier tails when compared to GCNTF-3. For compressor modelling the results show improvements when using TFiLM only for long release values, with GCNTF-3 performing best. Conversely, for both 50 ms and 250 ms release time, LSTM96 shows the lowest error. To understand how the models performance differ, we show the error distribution in Fig. 6. Also, by looking at the difference between L1 and MR-STFT error across time it is possible to identify a divergence around 1.50 minutes, of which we show an excerpt.

**Other effects —** Results for Face Bender, LA-2A, and MCompressor are shown in Table 1. In almost all cases the use of time-varying feature modulation results in lower overall error, showing a generalised improvement with respect to the state-of-the-art.

**Channel width —** Table 2 demonstrates how increasing the number of the parameters of the base GCN model does not lead to a performance improvement. This indicates that is not simply the increased number of parameters in TFiLM models that leads to an improvement, but instead time-varying modulation of activations.

**Block size —** Results for our proposed method at different block sizes are reported in Table 3. For GCNTF-3 trained on Custom Fuzz, there seems to be an optimal block size of $B = 128$ samples.

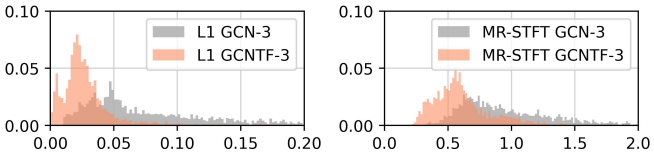


**Fig. 5**: GCN-3 and GCNTF-3 modelling fuzz with 2500 ms release

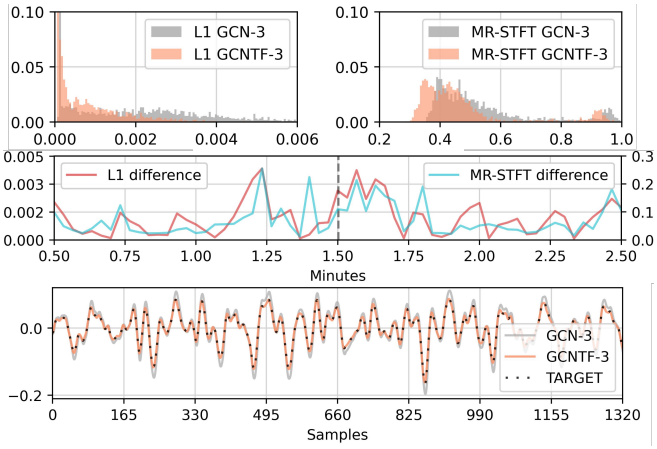


**Fig. 6**: GCN-3 and GCNTF-3 for compressor with 2500 ms release. Top: histogram of L1 and MR-STFT error. Middle: difference between error for the two models. Bottom: waveforms at 1.50 min

## 5. CONCLUSION

In this work, we presented a method for black-box modelling of audio effects with long-range dependencies by integrating time-varying feature-wise modulation into state-of-the-art convolutional models. We demonstrated that current state-of-the-art approaches fail to model behaviours over long time-scales for effects like fuzz and compressor, while our proposed method successfully captures them without increasing the receptive field of the processing network. These results open up future work to extend the approach to time-varying effects like chorus or tremolo, but also in applications like the proxy network approach for learning to control effects. Datasets, source code, and pretrained models are openly provided.

## 6. ACKNOWLEDGEMENTS

Funded by UKRI and EPSRC as part of the "UKRI CDT in Artificial Intelligence and Music", under grant EP/S022694/1.

## 7. REFERENCES


[1] Thomas Wilmering, David Moffat, Alessia Milo, and Mark B. Sandler, "A history of audio effects," *Applied Sciences*, vol. 10, no. 3, 2020.

[2] Matti Karjalainen and Jyri Pakarinen, "Wave digital simulation of a vacuum-tube amplifier," in *ICASSP*, 2006.

[3] David T Yeh, Jonathan S Abel, and Julius O Smith, "Automated physical modeling of nonlinear audio circuits for real-time audio effects—Part I: Theoretical development," *TASLP*, vol. 18, no. 4, pp. 728–737, 2009.

[4] Felix Eichas, Stephan Möller, and Udo Zölzer, "Block-oriented modeling of distortion audio effects using iterative minimization," in *DAFx*, 2015.

[5] Felix Eichas, Etienne Gerat, and Udo Zölzer, "Virtual analog modeling of dynamic range compression systems," in *142nd Convention of the Audio Engineering Society (AES)*, 2017.

[6] Etienne Gerat, Felix Eichas, and Udo Zölzer, "Virtual analog modeling of a UREI 1176LN dynamic range control system," in *143rd Convention of the Audio Engineering Society (AES)*, 2017.

[7] Fabián Esqueda, Boris Kuznetsov, and Julian D Parker, "Differentiable white-box virtual analog modeling," in *DAFx*, 2021, pp. 41–48.

[8] Julian D Parker, Sebastian J Schlecht, Rudolf Rabenstein, and Maximilian Schäfer, "Physical modeling using recurrent neural networks with fast convolutional layers," *arXiv preprint arXiv:2204.10125*, 2022.

[9] Joseph T. Colonel, Marco Comunità, and Joshua Reiss, "Reverse engineering memoryless distortion effects with differentiable waveshaper," in *153rd Convention on the Audio Engineering Society (AES)*. Audio Engineering Society, 2022.

[10] Shahan Nercessian, Andy Sarroff, and Kurt James Werner, "Lightweight and interpretable neural modeling of an audio distortion effect using hyperconditioned differentiable biquads," in *ICASSP*, 2021.

[11] John Covert and David L Livingston, "A vacuum-tube guitar amplifier model using a recurrent neural network," in *IEEE SoutheastCon*, 2013.

[12] Marco A Martínez Ramírez, Emmanouil Benetos, and Joshua D Reiss, "Deep learning for black-box modeling of audio effects," *Applied Sciences*, vol. 10, no. 2, pp. 638, 2020.

[13] Julian D Parker, Fabián Esqueda, and André Bergner, "Modelling of nonlinear state-space systems using a deep neural network," in *DAFx*, 2019.

[14] Christian J. Steinmetz and Joshua D. Reiss, "Efficient neural networks for real-time modeling of analog dynamic range compression," in *152nd Convention of the Audio Engineering Society (AES)*, 2022.

[15] Eero-Pekka Damskägg, Lauri Juvela, Etienne Thuillier, and Vesa Välimäki, "Deep learning for tube amplifier emulation," in *ICASSP*, 2019.

[16] Eero-Pekka Damskägg, Lauri Juvela, Vesa Välimäki, et al., "Real-time modeling of audio distortion circuits with deep learning," in *Sound and Music Computing Conf. (SMC)*, 2019.

[17] Thomas Schmitz and Jean-Jacques Embrechts, "Nonlinear real-time emulation of a tube amplifier with a long short time memory neural-network," in *144th Convention of the Audio Engineering Society (AES)*, 2018.

[18] Alec Wright, Eero-Pekka Damskägg, Vesa Välimäki, et al., "Real-time black-box modelling with recurrent neural networks," in *DAFx*, 2019.

[19] Alec Wright, Eero-Pekka Damskägg, Lauri Juvela, and Vesa Välimäki, "Real-time guitar amplifier emulation with deep learning," *Applied Sciences*, vol. 10, no. 3, pp. 766, 2020.

[20] Marco A Martínez Ramírez, Emmanouil Benetos, and Joshua D Reiss, "A general-purpose deep learning approach to model time-varying audio effects," in *DAFx*, 2019.

[21] Scott Hawley, Benjamin Colburn, and Stylianos Ioannis Mimilakis, "Profiling audio compressors with deep neural networks," in *147th Convention of the Audio Engineering Society (AES)*, 2019.

[22] Sawyer Birnbaum, Volodymyr Kuleshov, Zayd Enam, Pang Wei W Koh, and Stefano Ermon, "Temporal FiLM: Capturing long-range sequence dependencies with feature-wise modulations.," in *NeurIPS*, 2019.

[23] Dario Rethage, Jordi Pons, and Xavier Serra, "A WaveNet for speech denoising," in *ICASSP*, 2018.

[24] Qiao Tian et al., "TFGAN: Time and frequency domain based generative adversarial network for high-fidelity speech synthesis," *arXiv:2011.12206*, 2020.

[25] Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie, "Multi-band MelGAN: Faster waveform generation for high-quality text-to-speech," *arXiv:2005.05106*, 2020.

[26] Ethan Perez et al., "FiLM: Visual reasoning with a general conditioning layer," in *AAAI Conf. on Artificial Intelligence*, 2018.

[27] David Ha, Andrew Dai, and Quoc V Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[28] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu, "Dynamic convolution: Attention over convolution kernels," in *CVPR*, 2020.

[29] Christian Kehling, Jakob Abeßer, Christian Dittmar, and Gerald Schuller, "Automatic tablature transcription of electric guitar recordings by estimation of score-and instrument-related parameters.," in *DAFx*, 2014.

[30] Christian J. Steinmetz and Joshua D. Reiss, "auraloss: Audio focused loss functions in PyTorch," in *DMRN+15*, 2020.

[31] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP*, 2020.

[32] Christian J. Steinmetz, Jordi Pons, Santiago Pascual, and Joan Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," in *ICASSP*, 2021.