MODELING ANALOG DYNAMIC RANGE COMPRESSORS USING DEEP LEARNING AND STATE-SPACE MODELS

Hanzhi Yin[1,2], *Gang Cheng*[2], *Christian J. Steinmetz*[3], *Ruibin Yuan*[2], *Richard M. Stern*[1], *Roger B. Dannenberg*[1]

[1] School of Music, Carnegie Mellon University
[2] Multimodal Art Projection Research Community
[3] Centre for Digital Music, Queen Mary University of London

ABSTRACT

We describe a novel approach for developing realistic digital models of dynamic range compressors for digital audio production by analyzing their analog prototypes. While realistic digital dynamic compressors are potentially useful for many applications, the design process is challenging because the compressors operate nonlinearly over long time scales. Our approach is based on the structured state space sequence model (S4), as implementing the state-space model (SSM) has proven to be efficient at learning long-range dependencies and is promising for modeling dynamic range compressors. We present in this paper a deep learning model with S4 layers to model the Teletronix LA-2A analog dynamic range compressor. The model is causal, executes efficiently in real time, and achieves roughly the same quality as previous deep-learning models but with fewer parameters.

Index Terms— Virtual analog modeling; State-space model; Dynamic range compressor.

1. INTRODUCTION

Virtual analog modeling (VA modeling) concerns the digital simulation of analog audio devices like synthesizers and audio effect units [1, 2]. There has been a trend of using deep learning (DL) techniques in VA modeling, transforming VA modeling tasks to be data-driven by utilizing input and output waveform pairs processed by the analog system. Introducing VA modeling to DL may bring some research benefits. While these approaches can be used to emulate analog audio systems, they can also be used to construct differentiable proxies [3], facilitating tasks such as automatic mixing [4].

Until now, applications of DL to VA modeling have focused mostly on vacuum-tube amplifiers [5, 6] and distortion pedals [7, 8]. In contrast, dynamic range compressors (DRCs), which are non-linear, time-invariant, and possess longer temporal dependencies, have received less attention. Existing attempts include the use of autoencoders [9], using temporal convolutional networks (TCNs) [10], and gray-box models based on DRC implementation structures [11]. While these attempts investigated various aspects of the topic, there is still room to improve the performance. The output generated by some models still exhibits artifacts, and some best-performing models either rely on hard-coded components, are non-causal, or require a larger number of neural network parameters. The need for a model with greater objective accuracy and perceptual quality that is causal, parameter efficient, and real-time capable remains.

In 2021, Gu et al. proposed S4, which implements an infinite impulse response (IIR) in the state-space form for long-sequence modeling [12]. It has proven powerful because it can have an arbitrarily long receptive field. It can also preserve state information between samples or buffers to process arbitrarily long sequences. Because of this, S4 seems promising for improving a model's performance in modeling DRCs.

This work proposes a model that uses S4 layers to characterize an analog DRC, namely the Teletronix LA-2A compressor. This work introduced SSM to model an analog non-linear audio effect, exploring the effectiveness of using SSM in VA modeling. Various experiments are conducted to evaluate our model's objective and subjective performance and real-time evaluation capabilities. The proposed model provides roughly the same quality as previous deep-learning models but with a causal formulation and fewer parameters. It can also perform real-time inference given a specific audio buffer size, which is feasible in an audio production scenario.

2. BACKGROUND

2.1. Structured State Space Sequence Model (S4)

S4 is a neural network layer that implements an IIR system in state-space form. An N^{th}-order discrete SSM mapping of a mono signal to another mono signal can be expressed as follows, where u is the input signal, x is the intermediate signal, y is the output signal, and $\mathbf{A}(N \times N), \mathbf{B}(N \times 1), \mathbf{C}(1 \times N), \mathbf{D}(1 \times 1)$ are state-space matrices expressing linear mappings.

$$x[t] = \mathbf{A}x[t-1] + \mathbf{B}u[t] \tag{1}$$
$$y[t] = \mathbf{C}x[t] + \mathbf{D}u[t] \tag{2}$$

Finite impulse response (FIR) systems like convolutional neural networks (CNNs) have finite-length impulse responses. Since S4 is an IIR system, its impulse response is infinite, and its impulse response decay and receptive field are arbitrarily long. S4 layers are also parameter efficient, given that for filters with a similar effect, IIR systems require fewer parameters. There is an even more parameter-efficient S4 variant called S4D, with a diagonalized matrix $\mathbf{A}$ [13].

Inside an S4 layer, SSM matrices are complex-valued, allowing them to efficiently generate an impulse response that is as long as the input sequence using some mathematical techniques. The input sequence is filtered using Fast Fourier transforms, producing the output sequence and state information. S4 can process data sequences with state information preserved. It can calculate the internal state at the end of one segment, allowing the next buffer to be computed without discontinuities. For a very long sequence, one can input the entire sequence directly or section the sequences into small buffers and pass the state information. The entire recursive process is causal.

2.2. Feature-wise Linear Modulation

Analog audio devices usually feature external controls that modify these devices' operation. Digital emulations should also capture this behavior. To model DRC controls such as gain reduction, feature-wise linear modulation (FiLM) layers can be used [14], following the approach of some other VA modeling research [10]. Given an external information vector, FiLM first converts it into two vectors, γ and β, using a multi-layer perceptron (MLP). The output y is given by $y = \gamma \odot x + \beta$, where x is the input and $\odot$ is element-wise vector multiplication. FiLM layers enable adaptation of the model's behavior as a function of external controls.

3. METHODS

3.1. Proposed Model

Our S4-based model is illustrated in Fig. 1. At first, the input is expanded into c data channels. Unlike how CNN layers are conventionally implemented in DL, S4 layers are not combined or mixed across data channels. Instead, every linear layer applies affine transformations on the data's channel dimension, thus creating a mix or combination of channel data on a frame-by-frame basis.

The design is essentially a chain of S4 blocks, illustrated at the right of Fig. 1. Each block consists of a linear layer that mixes audio channels from c to c, a PReLU layer to introduce non-linearity, an S4D layer, a BatchNorm1D layer, a FiLM layer to introduce audio effect controls, another PReLU layer, and a residual connection. All S4 layers are S4D layers [13]. The BatchNorm1D layers treat the channel dimension as the feature dimension. An independent MLP processes external audio effect controls once to create control-based information

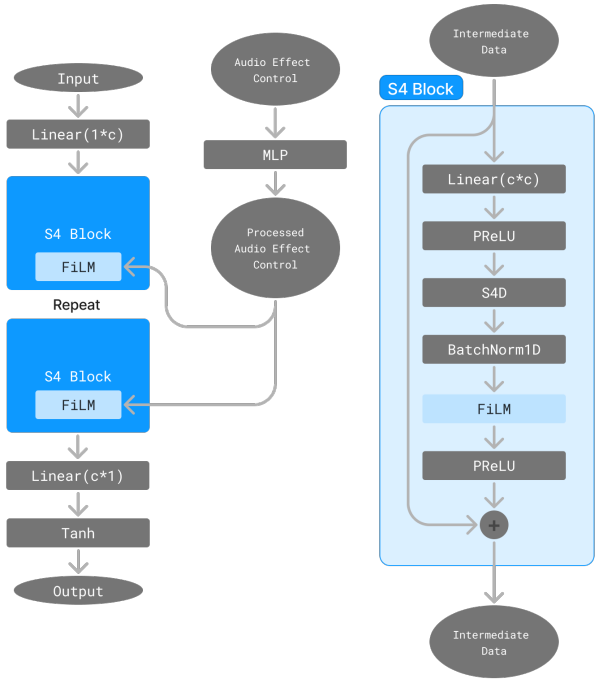

Fig. 1: The proposed S4 model. It mainly comprises a stack of S4 blocks, where the S4 layer models the temporal dimension, the linear layer models the channel dimension, the FiLM layer applies external controls, and PReLU layers apply non-linearities.

before processing audio. This control information is fed to all FiLM layers identically. Each FiLM layer has its own independent MLP to convert control information to γ and β and apply them to the data's channel dimension.

After the stack of S4 blocks, the final linear layer contracts audio channels from c to 1. The final $\tanh$ layer softly limits the output data to ± 1.0.

3.2. Experiment

We tested our model with four S4 blocks, 16 or 32 inner audio channels, and fourth or eighth S4D SSM order. There are four configurations in total. Models are named in the format of `ssm-c*-f*`, where `c` means the inner audio channel number and `f` means S4 SSM order. Our training code and audio samples are available online [1].

The SignalTrain dataset [15] is used to train, validate, and test those models. The SignalTrain dataset comprises audio input and output data processed by the Teletronix LA-2A compressor (LA-2A) with different gain reductions and compressing/limiting switch values. All audio data are mono sampled at 44.1 kHz. There are 87 540 s training data. Phase inversion is the only data augmentation technique applied with a probability of 0.5. To accommodate the SignalTrain dataset,

[1] https://int0thewind.github.io/s4drc/

the model takes audio waveforms and audio effect controls as 32-bit floating point vectors. The input audio is mono with an amplitude range of ± 1.0.

Models are trained using the SignalTrain dataset training split with batch size 32 in 60 epochs. The training audio data are segmented into sections of length 65 536 ($\approx$1.598 s at 44.1 kHz). No state information is preserved between audio buffers. Each buffer is processed independently. The learning rate is 0.001 and is reduced by a factor of 10 after ten epochs of no improvement in validation loss. The optimizer is AdamW, with default function arguments from `PyTorch` (v2.0.0). All S4D layer parameters' weight decays are set to zero. The training loss function combines both time and frequency domain loss and is the sum of mean-averaged error (MAE) with multi-resolution STFT (multi-STFT) loss [16], with default function arguments from `auraloss` (v0.4.0) [17]. Both parts are weighted equally.

Models are tested using the SignalTrain dataset testing split. The testing audio data are segmented with length 2^{23} ($\approx$190.218 s at 44.1 kHz) to test the model's long-term generalizability. The entire buffer is fed into the model without slicing it. When testing, MAE, mean-squared error (MSE), error-to-signal ratio (ESR) loss with a pre-emphasis filter of $H(z) = 1 - 0.85z^{-1}$ plus DC loss (ESR+DC) [18], multi-STFT loss with the same configuration in testing, loudness unit full scale (LUFS) difference with the ITU-R BS.1770 perceptual loudness recommendation, and Fréchet Audio Distance (FAD) [19] are evaluated. MAE, MSE, and ESR+DC loss are time-domain criteria, and multi-STFT loss is a frequency-domain criterion. ESR+DC loss may reflect the audio perceptual difference in the time domain, LUFS provides the loudness error, and FAD models perceptual similarity. We took Steinmetz and Reiss' TCN and LSTM models [10] as the baseline, as S4 is closely related to TCN, and our model structure and training procedure are close to them.

4. RESULT AND ANALYSIS

4.1. Objective Loss

The test losses of our models and various baseline models are presented in Table 1. Our `ssm-c32-f8` model has the best multi-STFT loss, and `ssm-c16-f8` has the best LUFS difference. Other best metrics are from `TCN-300-N` and `LSTM-32`, yet we found that our `ssm-c32-f4` model has very close MAE and MSE, and our `ssm-c32-f8` has very close FAD.

We found the time domain losses of the `ssm-c16-f8` and `ssm-c32-f8` models to be greater than those of our other models. Given that those higher loss values are from models with higher SSM filter orders, training a higher-order SSM might be more mathematically complicated.

We believe that our `ssm-c32-f4` model's performance is the most balanced. The `ssm-c32-f4` model has the best time-domain losses among all our models and outperforms all causal TCN models in all metrics. It provides MAE and MSE performances that are close to those of `TCN-300-N`, which uses three times more model parameters than `ssm-c32-f4` and is not causal. It also has a close FAD performance compared to `LSTM-32`, which cannot perform in real time and has higher time-domain and frequency-domain loss. Although the `ssm-c32-f8` and `ssm-c16-f8` models have better multi-STFT loss and LUFS, their time-domain loss is much higher than that of the `ssm-c32-f4` model. The `ssm-c32-f4` model has relatively good objective accuracy that is causal, parameter efficient, and real-time capable.

4.2. Subjective Listening Study

A multi-stimulus listening test similar to MUSHRA [20] was conducted to further evaluate the model's performance. The testing interface is webMUSHRA [21]. It allows online assessment, and participants can instantaneously switch between clips to facilitate the comparison of minute differences. Test participants score each audio clip based on the similarity to the reference and the effectiveness of the clip capturing the DRC's characteristics, with the range from 0 to 100; the higher, the better.

Eleven passages from the SignalTrain dataset testing split were included in the test, including strings, piano, guitar, and band clips. Each audio clip has 10 seconds. Three models were tested: `TCN-300-C` and `LSTM-32` from Steinmetz and Reiss [10] and our `ssm-c32-f4`. We also include the original output clip in the test as a reference. There are 17 valid responses. The results are illustrated in Fig. 2.

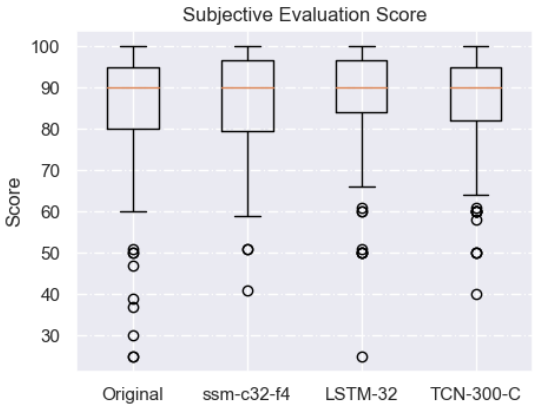

Fig. 2: Subjective Evaluation Scores among all Clips

Generally, we found no immediately distinguishable results from the three models we tested. All models show relatively the same median subjective scores and the same subjective score lower bounds, with our `ssm-c32-f4` model showing fewer outliers. As most participants demonstrated, it is hard to distinguish differences between testing clips, mak-

Model	Params	MAE	MSE	ESR+DC	Multi-STFT	LUFS	FAD
ssm-c16-f4	8.2k	1.012E-02	3.206E-04	4.003E-01	6.160E-01	6.249E-01	4.813E-02
ssm-c16-f8	9.3k	1.142E-01	2.432E-02	3.685E+00	6.588E-01	**3.518E-01**	4.318E-02
ssm-c32-f4	16.9k	*8.737E-03*	*2.879E-04*	*4.065E-01*	4.881E-01	4.766E-01	3.921E-02
ssm-c32-f8	18.9k	1.157E-01	2.503E-02	3.676E+00	**4.785E-01**	4.502E-01	*3.713E-02*
TCN-100-N	26k	1.580E-02	5.580E-04	2.331E-01	7.980E-01	1.155E+00	1.599E+00
TCN-300-N	51k	**7.660E-03**	**1.350E-04**	**2.913E-02**	6.160E-01	6.020E-01	1.062E-01
TCN-1000-N	33k	1.200E-01	2.650E-02	-	7.690E-01	9.340E-01	1.762E+00
TCN-100-C	26k	1.920E-02	1.390E-03	1.880E+00	7.840E-01	1.225E+00	1.903E+00
TCN-300-C	51k	1.440E-02	1.140E-03	1.800E+00	6.200E-01	7.610E-01	1.036E-01
TCN-1000-C	33k	1.170E-01	2.570E-02	3.150E+00	7.100E-01	8.990E-01	1.959E+00
LSTM-32	5k	1.100E-01	2.290E-02	1.870E+00	*5.650E-01*	*3.610E-01*	**2.741E-02**

Table 1: Our model's testing metrics and baseline models metrics from Steinmetz and Reiss [10]. Our models are on the top. `c` means inner audio channel number. `f` means S4 SSM order. For TCN models, `N` means non-causal. `C` means causal. Bold numbers are the global best metrics for all models. Italic numbers are the local best.

ing the scoring hard. To conclude, there appears to be no significant difference in the ratings between our and baseline models. The subjective listening study demonstrates a relatively close model performance between our S4 models and the previous TCN and LSTM models.

4.3. Real-time Performance

A real-time implementation requires that audio be processed incrementally in buffers to achieve a finite latency and that buffer computation time is less than the playback time. To evaluate computation time, we processed multiple buffers using six sizes of 128 through 4096 samples with an Apple M1 Max CPU core (no GPU) and the `ssm-c32-f4` model (44.1 kHz sampling rate), with state-passing. Note that S4 layers can preserve state from one block to the next, eliminating discontinuities at block transitions.

We define "speed ratio" as the audio playback time divided by the buffer stream inference time in `PyTorch`'s inference mode. A speed ratio higher than 1.0 means the inference speed is faster than the audio playback speed. The speed ratio on those buffer streams is presented in Fig. 3.

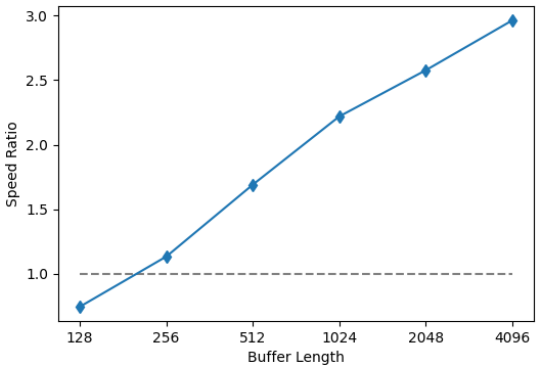

Fig. 3: Speed Ratios in Different Buffer Lengths.

The result shows that when the buffer size is greater than 256, the inference speed is faster than real time. Our implementation runs around three times as fast as real time using 4096 sample buffers. This shows that our model can perform in real time and that the implementation is feasible in an audio production scenario.

Our S4 implementation produces impulse responses to process audio buffer-by-buffer rather than implementing SSM directly to process audio sample-by-sample. The current S4 implementation might not portray the fastest real-time performance possible. The learned compressor could also be implemented by applying the state space updates sample-by-sample, reducing latency. We estimate this approach requires around 10M FLOPs per sample, so a single core capable of 5 GFLOPS should run faster than real time and perhaps even faster than the block-based approach.

5. CONCLUSION

We presented a model that uses S4 layers to model an analog DRC. Our model's objective performance and subjective performance are close to those of previous models but with a causal formulation and smaller model parameter space. Specifically, our model can perform in real time on a CPU core with a buffer size greater than 256, which is easily feasible in an audio production scenario. We showed that a model with SSM can efficiently emulate an analog non-linear audio effect with long temporal dependencies, ensuring causality and a small parameter space. While, in principle, SSMs process data sequences sample by sample, the current S4 implementation can only process data buffer by buffer. A promising goal of future work could be the utilization of state-space matrices to process audio data sample-by-sample to evaluate the fastest real-time performance.

References

[1] Jussi Pekonen and Vesa Välimäki, "The brief history of virtual analog synthesis," in *Proc. 6th Forum Acusticum. Aalborg, Denmark: European Acoustics Association*, 2011, pp. 461–466.

[2] Kurt James Werner, "Virtual Analog Modeling of Audio Circuitry using Wave Digital Filters," 2016.

[3] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts, "DDSP: Differentiable Digital Signal Processing," *8th International Conference on Learning Representations, ICLR 2020*, 1 2020.

[4] Christian J. Steinmetz, Jordi Pons, Santiago Pascual, and Joan Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2021-June, pp. 71–75, 2021.

[5] Alec Wright, Eero-Pekka Damskägg, Vesa Välimäki, and others, "Real-time black-box modelling with recurrent neural networks," in *22nd international conference on digital audio effects (DAFx-19)*, 2019, pp. 1–8.

[6] Eero Pekka Damskagg, Lauri Juvela, Etienne Thuillier, and Vesa Valimaki, "Deep Learning for Tube Amplifier Emulation," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May, pp. 471–475, 5 2019.

[7] Alec Wright, Eero Pekka Damskägg, Lauri Juvela, and Vesa Välimäki, "Real-Time Guitar Amplifier Emulation with Deep Learning," *Applied Sciences 2020, Vol. 10, Page 766*, vol. 10, no. 3, pp. 766, 1 2020.

[8] Eero-Pekka Damskägg, Lauri Juvela, Vesa Välimäki, and others, "Real-time modeling of audio distortion circuits with deep learning," in *Proc. Int. Sound and Music Computing Conf.(SMC-19), Malaga, Spain*, 2019, pp. 332–339.

[9] Scott H. Hawley, Benjamin Colburn, and Stylianos I. Mimilakis, "SignalTrain: Profiling Audio Compressors with Deep Neural Networks," 5 2019.

[10] Christian J. Steinmetz and Joshua D. Reiss, "Efficient neural networks for real-time modeling of analog dynamic range compression," *AES Europe Spring 2022 - 152nd Audio Engineering Society Convention 2022*, pp. 451–459, 2 2021.

[11] Alec Wright, Vesa Välimäki, and others, "Grey-box modelling of dynamic range compression," in *Proc. Int. Conf. Digital Audio Effects (DAFX), Vienna, Austria*, 2022, pp. 304–311.

[12] Albert Gu, Karan Goel, and Christopher Ré, "Efficiently Modeling Long Sequences with Structured State Spaces," 10 2021.

[13] Ankit Gupta, Albert Gu, and Jonathan Berant, "Diagonal State Spaces are as Effective as Structured State Spaces," 3 2022.

[14] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville, "FiLM: Visual Reasoning with a General Conditioning Layer," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 3942–3951, 4 2018.

[15] Benjamin Colburn and Scott Hawley, "SignalTrain LA2A Dataset," 5 2020.

[16] Ryuichi Yamamoto, Eunwoo Song, and Jae Min Kim, "Parallel Wavegan: A Fast Waveform Generation Model Based on Generative Adversarial Networks with Multi-Resolution Spectrogram," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2020-May, pp. 6199–6203, 5 2020.

[17] Christian J Steinmetz and Joshua D Reiss, "auraloss: Audio focused loss functions in PyTorch," in *Digital music research network one-day workshop (DMRN+ 15)*, 2020.

[18] Alec Wright and Vesa Valimaki, "Perceptual loss function for neural modeling of audio systems," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2020-May, pp. 251–255, 5 2020.

[19] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi Google, "Fr\'echet Audio Distance: A Metric for Evaluating Music Enhancement Algorithms," 12 2018.

[20] B Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.

[21] Michael Schoeffler, Sarah Bartoschek, Fabian Robert Stöter, Marlene Roess, Susanne Westphal, Bernd Edler, and Jürgen Herre, "webMUSHRA — A Comprehensive Framework for Web-based Listening Tests," *Journal of Open Research Software*, vol. 6, no. 1, pp. 8, 2018.