Steerable discovery of neural audio effects

Christian J. Steinmetz
c.j.steinmetz@qmul.ac.uk

Joshua D. Reiss
joshua.reiss@qmul.ac.uk

Centre for Digital Music, Queen Mary University of London, London, UK

Abstract

Applications of deep learning for audio effects often focus on modeling analog effects or learning to control effects to emulate a trained audio engineer. However, deep learning approaches also have the potential to expand creativity through neural audio effects that enable new sound transformations. While recent work demonstrated that neural networks with random weights produce compelling audio effects, control of these effects is limited and unintuitive. To address this, we introduce a method for the steerable discovery of neural audio effects. This method enables the design of effects using example recordings provided by the user. We demonstrate how this method produces an effect similar to the target effect, along with interesting inaccuracies, while also providing perceptually relevant controls.

1 Introduction

Audio effects are specialized signal processing tools used in music production for shaping the loudness, timbre, pitch, spatialization, or rhythm of sound [1]. There has been growing interest in deep learning for emulating analog audio effects [2–6], as well as methods for automatic control of audio effects to simplify music production [7–10]. In addition, related work in neural audio synthesis has investigated how deep learning may enable the synthesis of new sounds [11–14]. The expressive modeling capability of neural networks may enable audio effects that have not yet been realized by traditional signal processing approaches. However, deep learning methods for discovering new kinds of neural audio effects that expand or augment creative options for musicians remain limited.

Recent work demonstrated that neural networks with randomly initialized weights produce a range of compelling audio effects [15]. Furthermore, architectural details have an impact on the resulting effect, with deeper models leading to reverb and delay-like effects, and shallower networks leading to roomy distortion-like effects. However, this approach for generating neural audio effects is limited in two main ways. First, the resulting effects are largely dependent on the distributions from which the random weights are sampled. As a result, potentially interesting points in the weight space of these networks may never be reached. Second, while there is a connection between the architecture and the resulting effect, adjusting these attributes towards a sonic goal remains a challenging trial and error process. These challenges motivated our proposed method for a steerable generation process that allows users to direct the design of new effects according to their own aesthetic goals.

Most deep learning applications involve a clearly defined task, for example, emulating an audio effect as accurately as possible. However, the task of discovering new effects is clearly open-ended. In this case, we desire an approach that balances producing an "interesting" audio effect while ensuring this effect provides some level of intuitive (often perceptual) user control. We propose a simple steering method that uses a single, short input recording, along with a version of this recording processed by an audio effect similar to the one we would like to construct. We then train a conditional temporal convolutional network (TCN) [6] on this single input/output pair while holding the conditioning constant. After only a few minutes, this process produces a rough emulation of the target effect with some interesting inaccuracies characteristic of convolutional neural networks. Surprisingly, we find that adjusting the conditioning signal to values other than those seen during the steering process results in controls that tends to be correlated with perceptual attributes of the neural audio effect.

5th Workshop on Machine Learning for Creativity and Design at NeurIPS 2021, Sydney, Australia.

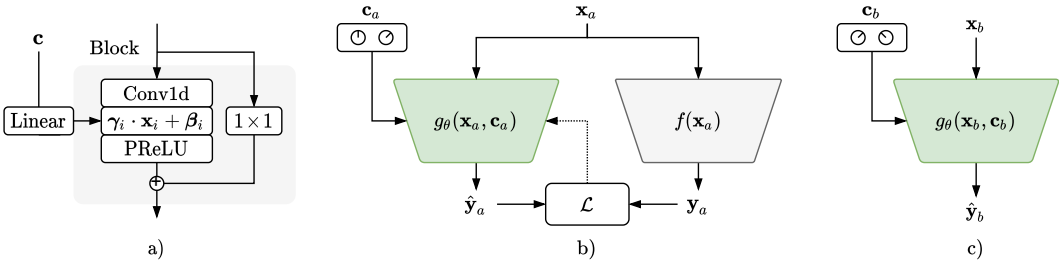


Figure 1: a) TCN block with 1D convolution, conditional affine transformation (FiLM), followed by a PReLU nonlinearity. b) Steering process where $g_\theta(\mathbf{x}_a, \mathbf{c}_a)$, a conditional TCN, is trained to emulate $f(\mathbf{x}_a)$, an existing audio effect, using a single input/output pair of recordings $\mathbf{x}_a, \mathbf{y}_a$. c) Generation process where $\mathbf{x}_b$, a new signal, is processed with the TCN and differing conditioning parameters $\mathbf{c}_b$.

2 Method

We begin by considering $g_\theta(\mathbf{x}_a, \mathbf{c}_a)$, a conditional TCN with weights $\theta \in \mathbb{R}^P$. This network processes an audio signal $\mathbf{x}_a \in \mathbb{R}^{C \times N}$ with C channels and N samples according to the conditioning signal $\mathbf{c}_a \in \mathbb{R}^D$ to produce a transformed signal $\hat{\mathbf{y}}_a \in \mathbb{R}^{C \times N}$. Our proposed steering process, shown in Figure 1b, requires an input signal $\mathbf{x}_a$ along with a version of this signal $\mathbf{y}_a$ that has been processed by another effect $f(\mathbf{x}_a)$, for example an existing reverb, dynamic range compressor, or other effect. We set the conditioning parameters $\mathbf{c}_a = \mathbf{0}$ and hold them constant during the steering process. Then we train the model using gradient descent, iteratively updating the weights $\theta := \theta - \eta \nabla_\theta \mathcal{L}\big(g_\theta(\mathbf{x}_a, \mathbf{c}_a), \mathbf{y}_a\big)$ with learning rate η, so the output $\hat{\mathbf{y}}_a$ is close to the processed signal $\mathbf{y}_a$ according to some loss function $\mathcal{L}$. After steering, we can use this network to process other audio signals, as shown in Figure 1c. Additionally, we can adjust the conditioning signal to another value $\mathbf{c}_b$, which will change the resulting effect by modulating the intermediate features in a different manner.

We utilize the formulation of the TCN presented in [6], which employs residual blocks as shown in Figure 1a. In order to integrate conditioning, we use feature-wise linear modulation (FiLM) [16]. To implement this, intermediate features at the i-th layer $\mathbf{x}_i$, with shape $B \times C \times N$ (batch, channels, samples), are modulated by a set of scaling $\boldsymbol{\gamma}_i$ and shifting $\boldsymbol{\beta}_i$ parameters. These parameters are derived from the global conditioning signal $\mathbf{c}$ which is projected from $\mathbb{R}^D$ to $\mathbb{R}^{2 \cdot C}$ at each intermediate layer. Finally, a PReLU [17] nonlinearity is applied. During optimization, we use Adam along with a formulation of the multiresolution STFT [18] for $\mathcal{L}$ based on the implementation in auraloss [19].

3 Results

We applied this steering process with examples from a compressor, analog delay, guitar amplifier, and reverberation effect. We used $\mathbf{c} \in \mathbb{R}^2$ to provide 2 control parameters and trained for 2500 steps using recordings at 44.1 kHz. We found it best to match the receptive field of the network to the expected memory of the effect. For example, we used a receptive field of ≈ 300 ms for the compressor and ≈ 2000 ms for reverberation. This process resulted in an effect with similar characteristics to the original effects, however with some inaccuracies such as interesting delay-like patterns and distortions. Furthermore, we found that varying the control parameters tends to change relevant aspects of the effect, such as the amount of compression or the decay time. We provide a Colab notebook[1] for these experiments along with listening examples that demonstrate the range of these effects.

To further investigate the perceptual relevance of the control parameters, we evaluate the model steered with the compressor signal by processing a piano recording, varying the conditioning $\mathbf{c}$ over a 2D grid from -5 to 5. We create a heatmap of the loudness in dB LUFS [20], as shown in Figure 2a, which demonstrates clear structure in the control parameter space. We perform a similar experiment for the reverberation effect, but instead use an impulse as input, measuring the T_{60}, shown in Figure 2b. Even though the TCN is nonlinear, we found the T_{60} provided a sense of the reverberation length, with the input level having a limited effect on the decay as shown in Figure 3.

These experiments demonstrate only a few applications of this approach. Future work may involve using steering signals originating from other audio effects, as well as unconventional sources, such as learning a mapping from one sound to another. The eventual goal would be to drive the steering process using only one signal, enabling the use of arbitrary recordings without the original clean signal. Other directions include experimentation with other architectures, as well as approaches for quantifying the "interestingness" of neural audio effects for use as an objective function.

[1] `https://csteinmetz1.github.io/steerable-nafx`

Acknowledgement

This work is supported by the EPSRC UKRI Centre for Doctoral Training in Artificial Intelligence and Music (EP/S022694/1).

References

[1] T. Wilmering, D. Moffat, A. Milo, and M. B. Sandler, "A history of audio effects," *Applied Sciences*, vol. 10, no. 3, 2020.

[2] E.-P. Damskägg, L. Juvela, V. Välimäki, *et al.*, "Real-time modeling of audio distortion circuits with deep learning," in *SMC Conference*, 2019.

[3] M. A. Martínez Ramírez and J. D. Reiss, "Modeling nonlinear audio effects with end-to-end deep neural networks," in *ICASSP*, 2019.

[4] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time guitar amplifier emulation with deep learning," *Applied Sciences*, vol. 10, no. 3, 2020.

[5] S. Hawley, B. Colburn, and S. I. Mimilakis, "Profiling audio compressors with deep neural networks," in *147th AES Convention*, 2019.

[6] C. J. Steinmetz and J. D. Reiss, "Efficient neural networks for real-time analog audio effect modeling," *arXiv:2102.06200*, 2021.

[7] D. Sheng and G. Fazekas, "A feature learning siamese model for intelligent control of the dynamic range compressor," in *IJCNN*, IEEE, 2019.

[8] S. I. Mimilakis, N. J. Bryan, and P. Smaragdis, "One-shot parametric audio production style transfer with application to frequency equalization," in *ICASSP*, IEEE, 2020.

[9] M. A. M. Ramírez, O. Wang, P. Smaragdis, and N. J. Bryan, "Differentiable signal processing with black-box audio effects," in *ICASSP*, 2021.

[10] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," in *ICASSP*, 2021.

[11] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *ICML*, 2017.

[12] L. Hantrakul, J. H. Engel, A. Roberts, and C. Gu, "Fast and flexible neural audio synthesis.," in *ISMIR*, 2019.

[13] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *ICLR*, 2020.

[14] B. Hayes, C. Saitis, and G. Fazekas, "Neural waveshaping synthesis," *ISMIR*, 2021.

[15] C. J. Steinmetz and J. D. Reiss, "Randomized overdrive neural networks," in *4th Workshop on Machine Learning for Creativity and Design at NeurIPS 2020*, 2020.

[16] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *AAAI*, 2018.

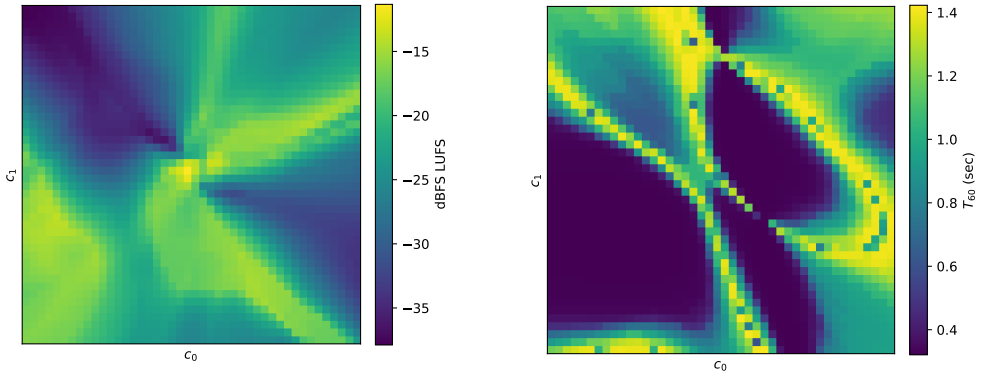
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, pp. 770–778, 2016.

[18] 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 *ICASSP*, 2020.

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

[20] "Algorithms to measure audio programme loudness and true-peak audio level," recommendation, International Telecommunications Union, October 2015.

Supplementary materials



a) Dynamic range compressor b) Artificial reverberation

Figure 2: Parameter space $\mathbf{c} \in \mathbb{R}^2$ from -5 to 5 with relation to a) loudness dB LUFS for a model steered with a signal from a dynamic range compressor, and b) T_{60} for a model steered with a signal from an artificial reverberation effect, both of which demonstrate clear structure.

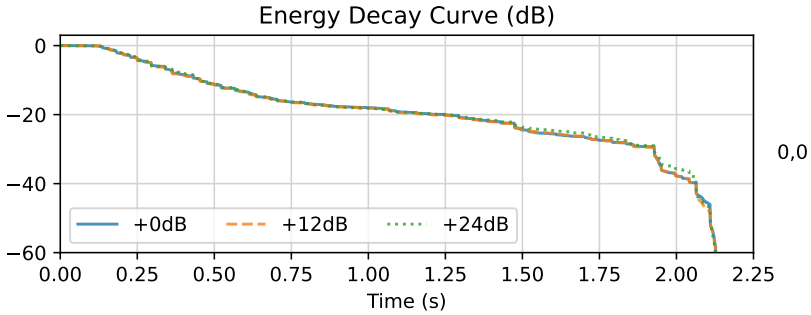


Figure 3: Multiple energy decay curves with an impulse at varying input levels as input to the model. Even though the TCN is nonlinear, the decay appears consistent as expected from a reverb-like effect.

	Models	
Parameter	Compressor	Reverberation
Layers	4	5
Channels	32	32
Kernel size	9	9
Dilation growth	10	10
Learnable parameters	21,803	32,268
Receptive field (samples)	8889	88889
(ms)	201.6	2015.6
Learning rate	1e-3, 1e-4 (80%), 1e-5 (95%)	
Optimizer	Adam	
Iterations	2500	
MR-STFT sizes	{32, 128, 512, 2048}	
Train time	4 min 26 sec	7 min 4 sec
MR-STFT Error	0.4945	0.9453

Table 1: Model hyperparameters