forum acusticum 2023

DIFFERENTIABLE MODELLING OF PERCUSSIVE AUDIO WITH TRANSIENT AND SPECTRAL SYNTHESIS

Jordie Shier[1*†] **Franco Caspe**[1†] **Andrew Robertson**[2]
Mark Sandler[1] **Charalampos Saitis**[1] **Andrew McPherson**[3]
[1] Centre for Digital Music, Queen Mary University of London, United Kingdom
[2] Ableton AG, Berlin, Germany
[3] Dyson School of Design Engineering, Imperial College London, United Kingdom

ABSTRACT

Differentiable digital signal processing (DDSP) techniques, including methods for audio synthesis, have gained attention in recent years and lend themselves to interpretability in the parameter space. However, current differentiable synthesis methods have not explicitly sought to model the transient portion of signals, which is important for percussive sounds. In this work, we present a unified synthesis framework aiming to address transient generation and percussive synthesis within a DDSP framework. To this end, we propose a model for percussive synthesis that builds on sinusoidal modeling synthesis and incorporates a modulated temporal convolutional network for transient generation. We use a modified sinusoidal peak picking algorithm to generate time-varying non-harmonic sinusoids and pair it with differentiable noise and transient encoders that are jointly trained to reconstruct drumset sounds. We compute a set of reconstruction metrics using a large dataset of acoustic and electronic percussion samples that show that our method leads to improved onset signal reconstruction for membranophone percussion instruments.

---

†FC and JS contributed equally and share first authorship
Corresponding author: *j.m.shier@qmul.ac.uk.*

1. INTRODUCTION

Modeling of instrumental tones using data-driven methods and neural networks has received considerable attention in recent years. Differentiable digital signal processing (DDSP) enables classical audio synthesis algorithms to be integrated with neural networks and incorporated within gradient descent training regimes. DDSP synthesis methods have enabled high-quality and controllable audio synthesis with less training data compared to other deep learning approaches by leveraging the signal generation capabilities of DSP [1]. Additionally, these techniques have enabled rich synthesizer interaction methods, including audio-driven control of synthesizers, interpretable controls based on time-varying pitch envelopes, and timbre transfer applications.

The first work on DDSP by Engel et al. [2] used a differentiable sinusoidal plus noise synthesizer based on Serra's original method [3] for instrumental audio synthesis. While a large body of research has followed, it has focused almost exclusively on modeling of harmonic tones with little application to unpitched percussion sounds. Furthermore, DDSP synthesis approaches have not sought to explicitly address the transient / onset portion of instrument signals, a signal component that is both important to musical audio perception [4] and is known to be poorly handled by sinusoidal plus noise synthesis [5]. To this end, we explore sinusoidal plus noise modeling synthesis for non-harmonic percussion signals (i.e., drums and cymbals) and propose the addition of a temporal convolutional network (TCN) to handle transient / onset signal components in a differentiable framework. Within our approach, a non-harmonic sinusoidal signal is generated using sinusoidal modeling synthesis (SMS) [3] and parame-

ter encoders for noise and transient synthesizers are jointly trained. A latent representation of the transient signal is generated by the transient encoder, paving the way for future work on controllable drum synthesis within a differentiable, data-driven paradigm.

In this work, we focus on the reconstruction of unpitched drum sounds to explore the strengths and weaknesses of our proposed method for drum synthesis. We train our model using a diverse dataset of electronic and acoustic drumset sounds and evaluate the results using a number of reconstruction metrics. We find that our proposed method improves onset reconstruction for membranophones, although adds artifacts that degrade the decay portion of the sound. We conclude with suggestions for future work focused on controllable differentiable drum synthesis, which builds on the methods proposed here.

2. BACKGROUND

2.1 Drum Synthesis

Synthesis of drum sounds has been explored extensively in previous work. In contrast to physically-informed approaches for percussion synthesis [6, 7], spectral synthesis methods, which include sinusoidal modeling synthesis (SMS) [3], seek to model perceptual qualities of audio signals. SMS has been used for percussive synthesis [8], including as an analysis step for modal synthesis [6]. Recent research used a time-frequency analysis method and sinusoidal synthesis to model tom drum sounds that were indistinguishable from real sounds in a listening test [9], pointing to the potential for SMS for high-quality, real-time drum synthesis. In contrast to traditional synthesis methods, data-driven approaches learn synthesis functions from a corpus of audio. Related work specific to drum synthesis has utilized convolutional neural networks [10], generative adversarial networks [11], and diffusion networks [12].

2.2 Differentiable Digital Signal Processing

Differentiable digital signal processing (DDSP) combines the strengths of traditional DSP approaches with data-driven approaches. A significant body of work has explored the application of DDSP for the synthesis of harmonic instrumental tones [2, 13], including pitched percussion (i.e., piano) [1]. Directly estimating the frequencies of oscillators using gradient descent is a challenging problem due to the oscillatory nature of loss surfaces produced by current audio reconstruction objectives [14, 15].

Most approaches rely on pitch estimation algorithms and harmonic oscillators to circumvent this problem. As a result, DDSP has seen limited application in the modeling of non-harmonic sounds, and therefore unpitched percussion instruments. The one exception is the recent work by Diaz et al. [16], which implemented a differentiable modal resonator. Here, we identify the lack of transient modeling as another limitation of current DDSP methods for synthesizing percussive sounds, and propose the use of TCNs to generate these signal components.

2.3 Transient Modeling

Transient and onset signals, defined by abrupt changes in amplitude, phase, or frequency information [17], are not well represented by sinusoidal plus noise models, which is the basis of many DDSP approaches. While musical onsets and transient regions are an important perceptual component for all musical sounds [4], they are particularly important for percussive audio, which often contains rapidly decaying signal components following an impulsive event. Levine and Smith [18] suggest that sinusoidal, noise, and transient signal components be modeled separately and proposed a method for isolating transients. Similar approaches proposed extracting attack or transient signal components [3], applying modifications to the sinusoidal signal components, and then re-inserting transients directly to the output. Verma and Meng [5] suggest that sinusoidal modeling can also be used to model transient signals if conducted in the correct signal domain, and propose doing so in the discrete cosine transform (DCT) domain. An alternative approach proposed a source filter method [17], reframing the transient modeling formulation as transient $*$ sines $+$ noise. Conceptually, our approach is similar to this source filter formulation; however, we use gradient descent to estimate parameters for a TCN which acts as a filter for transient generation.

2.4 Temporal Convolutional Networks

Temporal convolutional networks (TCNs) use multiple layers of time domain convolutions followed by non-linear activation functions and have been successfully applied to many deep learning audio tasks including audio synthesis [19] and audio effect modeling [20]. Dilated convolution kernels with exponentially increasing dilation rates enable larger receptive fields with relatively few layers, helping to address temporal signal dependencies. A dilated TCN was used by Wang et al. [21] in a source-filter based approach for voice synthesis. We explore a

similar approach for transient synthesis using TCNs and a sinusoidal input signal.

3. METHOD

Our proposed approach, shown in Figure 1, consists of three parallel analysis and synthesis pipelines that aim to model an input drum signal y using sinusoidal, noise, and transient components. Their outputs are then combined following a mixing strategy to resynthesize an approximation $\hat{y}$ of the input.

3.1 Sinusoidal Modeling

The sinusoidal pipeline seeks to model tonal signal components with a set of time-varying sinusoids. In particular, the system aims to extract parameters $\langle a_i^j, f_i^j, \phi^j \rangle$, which denote the ith frame-wise amplitude and frequency, and the initial phase of the jth sinusoidal component, respectively. Due to the aforementioned challenges related to estimating frequency parameters using gradient descent, we instead employ a well-known tracking algorithm based on sinusoidal peak picking [3]. We use the constant-Q transform (CQT) for our time-frequency representation as it provides improved frequency resolution, which is important for certain percussion sounds (e.g., kick drums). We track N_S sinusoidal components on the CQT spectrogram of the input signal, with a minimum frequency of 20 Hz, 24 bins per octave, and a total frequency span of 10 octaves. Finally, we synthesize the sinusoidal components S_n by linearly interpolating the amplitudes and frequencies from frame to sample rate and adding the extracted initial phase to each component, as shown in Equation 1.

$$S_n = \sum_{j=1}^{N_S} a_n^j sin(2\pi \cdot f_n^j \cdot n + \phi^j) \tag{1}$$

3.2 Noise Modeling

The noise pipeline consists of a frame-level analysis network implemented with a SoundStream encoder [22] without its residual vector quantizer and a filtered noise generator with linear frequency bands based on [2]. The encoder is composed of a 1D convolutional input layer, followed by a series of non-causal 1D convolutional blocks, each including three residual units of tripling dilation ratio, and a strided downsampling layer with ELU activation. Finally, an output convolutional layer projects the intermediate representation to the appropriate number of output channels.

The noise encoder takes the input audio and predicts a frame-wise sequence $\nu_i \in \mathbb{R}^{N_N}$ of gain coefficients for filters equally spaced on a linear frequency scale, with i denoting the ith frame and N_N the number of filter bands. Since our sinusoidal model presents high resolution on the low-frequency spectrum, we use a noise generator on a linear scale to account for the high frequencies of drum signals even when using a few noise bands.

Finally, the encoder's output is then relayed to the control input of the noise generator, which synthesizes an impulse response for each set of filter coefficients ν_i using the inverse discrete Fourier Transform, here denoted as $F^{-1}(.)$. The inversion yields an impulse response of length $2N_N$ which is then padded with zeroes left and right up to a length $2h$. They are convolved with a white noise signal ϵ_n and then aggregated using overlap-add over L Hanning windows $w(.)$ of length $2h$, with h being the hop size of the operation. The noise component N is generated as shown in Equation 2.

$$N_n = \sum_{i=1}^{L} w(n - hi)(\epsilon_n * F^{-1}(\nu_i)) \tag{2}$$

3.3 Transient Modeling

Given a drum signal resynthesized with sinusoidal and noise components as input, we design a pipeline to generate transients that are not effectively modeled by the other components. The transient modeling pipeline has two goals: i) learn a controllable non-linear transfer function $T : x, z \rightarrow y$ that processes the sinusoidal and noise components as monophonic audio signals of length L, $x \in \mathbb{R}^L$ and generates an output with improved transients $y \in \mathbb{R}^L$; ii) learn a controller network that predicts a transient conditioning vector $z \in \mathbb{R}^D$ to control transient synthesis for a particular input.

We implement the transfer function with a TCN architecture similar to [20], which can be seen as a chain of non-linearities and filters that operate at the audio rate. We use Feature-wise Linear Modulation (FiLM) [23] as a method to control the distortion and filtering characteristics of the TCN, conditioned on the encoded input drum signal z. FiLM functions by applying an affine transform to the output of the convolution of each layer of the TCN. A unique multi-layer perceptron is learned for each TCN layer and maps from the transient encoding z produced by the controller network to shift and scale parameters for the affine transform. The controller network is implemented with a SoundStream encoder similar to the noise encoder.
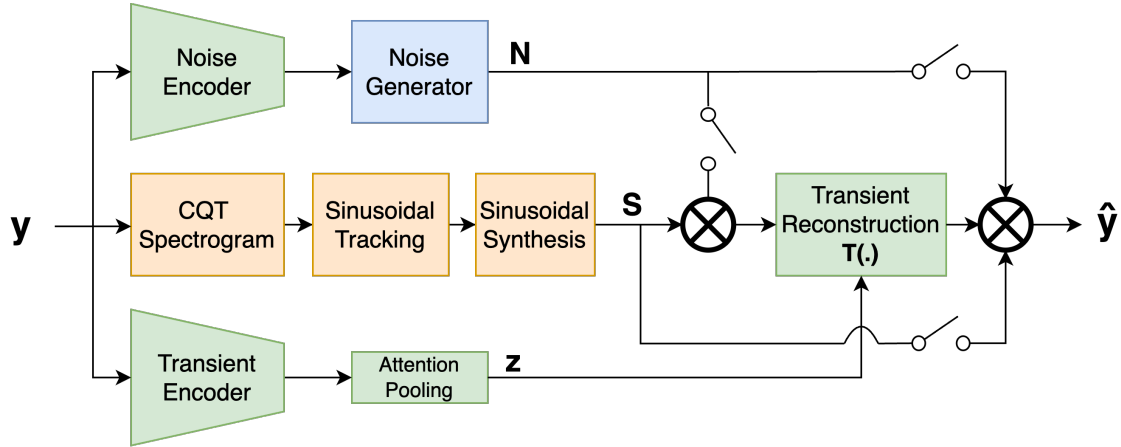
10th Convention of the European Acoustics Association
Turin, Italy • 11th – 15th September 2023 • Politecnico di Torino

2227

Figure 1. Our system models an input drum waveform y using three parallel pipelines: sinusoids, noise, and transients. The sinusoidal components are obtained by tracking, while noise and transient components are learned through differentiable modeling. A function, $T(.)$, is learned for reconstructing transients from the sinusoidal and noise components, which is controlled by a vector z generated by a transient encoder. We explore several strategies for mixing signal components, which are fed to $T(.)$ or directly to the output, as indicated by the switches in the figure. A mixing scheme is predefined for a model.

However, we use attention pooling to aggregate the framewise features predicted by the transient encoder into a single latent embedding vector z used for FiLM.

4. EXPERIMENTS

We conducted a series of experiments to evaluate our approach in terms of audio reconstruction. To this end, we composed several different configurations of our model to help us understand the potential benefits of using a TCN to generate transient signal components. We consider four different mixing strategies: 1) transient generation from sines only, denoted $T(S)$; 2) transient generation from mixed sines and noise, denoted $T(S + N)$; 3) transient generation from sines, with noise added in parallel, denoted $T(S) + N$; 4) transient generation from sines, with noise and sines added in parallel, denoted $T(S) + S + N$. We baseline these approaches against sinusoids only S and sinusoids plus noise only $S + N$. Audio examples of results and architecture details are provided on an accompanying website.[1]

Due to the lack of high-quality, open-source one-shot drum samples, we curated our experimental dataset

---

[1] https://jordieshier.com/projects/differentiable_transient_synthesis/

from the author's collection of commercial sample packs. While the FreeSound One-Shot Percussion dataset [10] contains over 10k sounds, the dataset sample rate is 16kHz and samples aren't annotated with sound source or instrument. We used samples that were professionally produced at full audio resolution, allowing us to conduct our experiments at a sample rate of 48kHz, which we feel is important for evaluating transient signal components and improves the applicability of our method to music production contexts.

The final dataset contained 25k samples with an equal split of acoustic and electronic sources, and included kick, snare, tom, hihat, cymbals, and a variety of other percussion instruments. All samples were preprocessed to remove starting silence and trimmed or padded to two seconds. Samples were distributed into train, validation, and testing splits (80/10/10), ensuring an equal balance of acoustic and electronic samples and that samples originating from the same sample pack (i.e., recorded on the same drum in the same room) were contained within a single split.

We configure the sinusoidal tracker with a maximum number of sinusoids N_S=64, and a hop size of 256 samples. The noise generator uses N_N=128 noise bands and a hop size h=128 samples. We use a transient conditioning

Table 1. Overall reconstruction error metrics. Under method, S, N, and $T()$ refer to the sinusoidal, noise, and transient synthesis methods, respectively. MSS is multi-resolution spectral loss (the train loss), LSD is log spectral distance, and SF is spectral flux onset error.

Method	All			Acoustic			Electronic		
	$MSS \downarrow$	$LSD \downarrow$	$SF \downarrow$	$MSS \downarrow$	$LSD \downarrow$	$SF \downarrow$	$MSS \downarrow$	$LSD \downarrow$	$SF \downarrow$
S	1.34	2.51	153.7	1.66	3.64	80.9	1.03	1.38	226.6
$S + N$	0.92	**1.20**	152.6	1.06	**1.65**	79.6	0.77	**0.75**	225.8
$T(S)$	0.99	1.70	131.1	1.16	2.31	54.9	0.81	1.08	207.5
$T(S + N)$	0.87	1.45	128.7	1.00	1.80	59.2	0.74	1.09	198.3
$T(S) + N$	**0.85**	1.36	128.8	**0.99**	1.74	**54.6**	**0.71**	0.99	203.1
$T(S) + S + N$	0.86	1.39	**123.4**	1.00	1.75	56.5	0.73	1.02	**190.1**

Table 2. Reconstruction metrics calculated on specific percussion instruments. Cymbals include hihats.

Method	Kick			Snare			Tom			Cymbals		
	$MSS \downarrow$	$LSD \downarrow$	$SF \downarrow$	$MSS \downarrow$	$LSD \downarrow$	$SF \downarrow$	$MSS \downarrow$	$LSD \downarrow$	$SF \downarrow$	$MSS \downarrow$	$LSD \downarrow$	$SF \downarrow$
S	1.22	2.18	470.5	1.37	2.44	84.8	1.31	3.10	185.1	1.47	2.64	12.8
$S + N$	0.92	**1.06**	470.4	0.91	**1.03**	84.0	0.81	**1.27**	185.0	**0.98**	**1.43**	**10.3**
$T(S)$	0.93	1.60	388.8	0.93	1.47	62.8	0.93	1.83	181.2	1.11	1.90	11.7
$T(S + N)$	0.79	1.34	**364.6**	0.83	1.30	58.3	0.77	1.47	200.1	1.00	1.62	12.7
$T(S) + N$	**0.76**	1.28	379.6	**0.81**	1.21	58.3	**0.75**	1.41	183.0	0.99	1.53	12.6
$T(S) + S + N$	0.77	1.30	372.1	0.84	1.24	**54.4**	0.76	1.43	**167.0**	1.01	1.56	11.4

vector of length 128: $z \in \mathbb{R}^{128}$ and the TCN is composed of 8 blocks with a dilation factor of 2 and 32 hidden channels. While the TCN can process monophonic audio signals of any duration, all samples used in these experiments had a duration of two seconds. All encoders were optimized using a multi-resolution spectral loss with the same configuration of FFT, window, and hop sizes as in [24], which has also been used in prior work using TCNs for audio effect modeling [20]. This loss is calculated as a weighted sum of the spectral convergence and log-magnitude spectral difference. We used an Adam optimizer with an initial learning rate of $1e^{-4}$ and a batch size of 12. Similar to [20], the learning rate was scheduled to decrease by a factor of two if the validation loss did not decrease for 20 epochs, and training was halted if validation loss failed to improve for a further 20 epochs. All training runs were capped at 48 hours.

4.1 Reconstruction Evaluation

We consider three metrics to evaluate audio reconstruction: multi-resolution STFT error (same as the training loss, denoted MSS), log spectral distance (LSD) using the same formulation as [25], and the mean absolute error between the spectral flux onset signals (SF) [26] extracted from y and $\hat{y}$. The spectral flux onset signal is calculated as the L_2 norm on the rectified spectral difference:

$$SF(n) = \sum_{\omega=0}^{\frac{N}{2}+1} H(|X_\omega(n)| - |X_\omega(n-1)|)^2 \quad (3)$$

where $H(x)$ is a rectified linear unit, $H(x) = (x+|x|)/2$, which includes only positive differences to emphasize the onset. Because applying a log transformation de-emphasizes transients, the LSD provides insight into the

10th Convention of the European Acoustics Association
Turin, Italy • 11th – 15th September 2023 • Politecnico di Torino
2229

reconstruction of the signal with the decay emphasized, whereas SF error provides insight into the reconstruction of signal transients and onsets.

Reconstruction results computed on acoustic and electronic percussion samples, separately and combined, are shown in Table 1. Additionally, reconstruction metrics are shown for a selection of individual percussion instruments: kicks, snares, toms, and cymbals (which includes hihat) in Table 2. Looking at the spectral flux onset error, the transient TCN models achieved the lowest error in all tests except for the individual cymbal sounds, where the sines plus noise model outperforms TCN models. The sines plus noise model achieved the best reconstruction results in terms of LSD for all tests. Interestingly, the MSS loss, which was the training objective, was lowest for transient TCN models, except for the individual cymbals. Investigating the separate spectral loss terms contributing to the MSS (i.e., spectral convergence and log-magnitude spectral difference), we note that models containing a TCN more efficiently decreased the spectral convergence term compared to the log-magnitude spectral difference.

These results point to the following four takeaways: 1) Using a transient TCN generally improved signal onset reconstruction for membranophones (i.e., kick, snare, tom); 2) Sinusoidal plus noise without the TCN worked the best for idiophones (i.e., cymbals and hihats); 3) Generally, models that use noise summed in parallel performed better than models without; and 4) The transient TCN more effectively reduced spectral convergence loss during training, which appears to be correlated with improved onset reconstruction. The implication is that TCNs improved onsets at the expense of the signal decay. This was verified in an informal listening test during which we noted the addition of artifacts, similar to comb filtering, in the decay portion of TCN reconstructions.

4.2 Visualizing the Transient Embedding

We visualized the test dataset by producing embeddings using the trained transient encoder to provide further insight into the transient parameter space. For each audio sample, the result is an embedding vector $z \in \mathbb{R}^{128}$. For visualization, we use t-SNE [27], a common dimensionality reduction technique used to visualize high-dimensional data, to map embeddings to a two-dimensional space. Figure 2 shows the resulting 2D mappings; samples are separately plotted for acoustic and electronic samples and are colour-coded by instrument type to highlight clusters

based on instruments within the embedding space. The embedding space features shared characteristics within instrument and sample types, showing small overlaps across these groups. This indicates that there is a continuum in the space, where in many cases, different instruments in a vicinity feature similar embeddings. However, there is a progressive variation across instrument and sample types, with transient characteristics becoming more salient at specific locations. For instance, the electronic and acoustic sounds clearly occupy separate areas of the point cloud. We additionally note the relatively large distance between kicks, hihats, and cymbals groups.

Observing the samples in the point cloud, we can infer that the embedding of our encoder represents well the sound characteristics required by the different samples. This opens up the possibility to parameterize the space to gain high-level control over the transients of drum sounds. This could serve as a first stepping stone towards controllable differentiable drum synthesis.

5. CONCLUSIONS

We presented a neural audio synthesis architecture for drum modeling in terms of three major sound components. In particular, we used a TCN modulated with FiLM embeddings as a method to reconstruct transient / onset signals within a sines plus noise synthesis model. We trained parameter encoders for transient and noise signal generators on a diverse dataset of acoustic and electronic drumset sounds and evaluated the resulting models using reconstruction metrics. Although we modelled transients indirectly using MSS loss, the TCN learned to model short-term signal features associated with membranophone onsets. We note that this came at the expense of added artifacts during the decay portion of the signal, which is reflected by the LSD. This points to the need for further work looking at how we can balance resynthesis fidelity of transient and decay signal components.

Our work takes a step towards a fully differentiable model for percussive synthesis using a neural source-filter approach for transient modeling. One major technical roadblock to realizing a fully differentiable percussion synthesizer is frequency estimation using gradient descent. Future work may leverage recent findings by Hayes et al. [15] for this problem. Beyond exploring solutions to these technical challenges, future work includes investigating the affordances of our analysis-synthesis framework for creative practices. For instance, by leveraging the latent space of a variational autoencoder for high-level
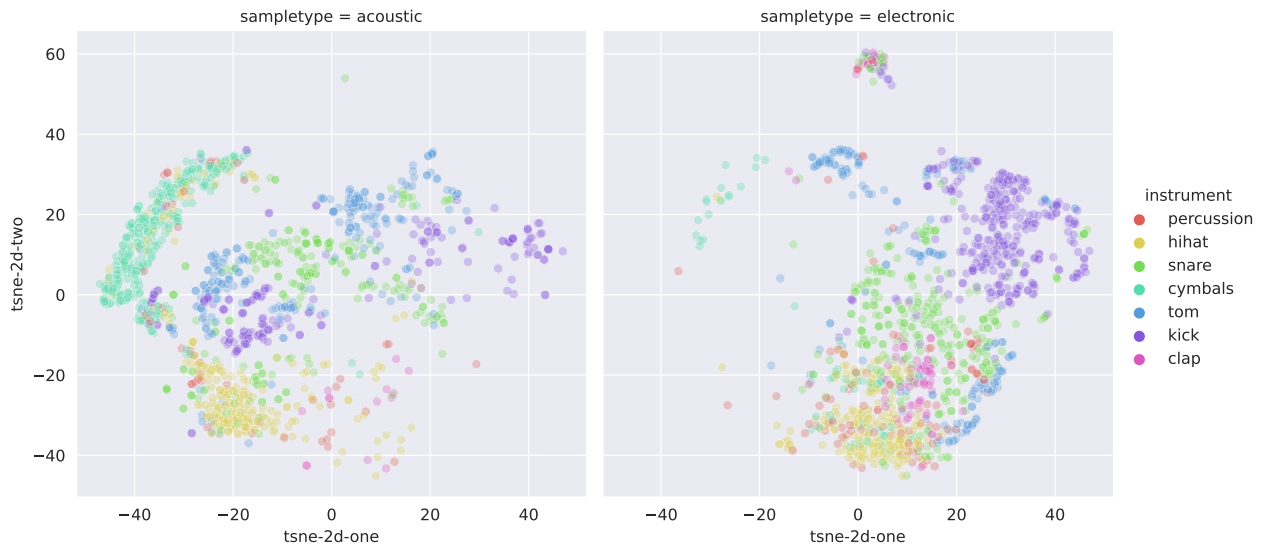
Figure 2. t-SNE plots of transient embeddings z colored by percusison type. Left plot is acoustic instruments and right plot is electronic.

control of each sonic component.

7. REFERENCES

[1] L. Renault, R. Mignot, and A. Roebel, "Differentiable Piano Model for Midi-to-Audio Performance Synthesis," in *Proc. of the 25th Int. Conf. on Digital Audio Effects*, 2022.

[2] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Int. Conf. on Learning Representations*, 2020.

[3] X. Serra and J. Smith, "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[4] K. Siedenburg, "Specifying the perceptual relevance of onset transients for musical instrument identification," *The Journal of the Acoustical Society of America*, vol. 145, no. 2, pp. 1078–1087, 2019.

[5] T. S. Verma and T. H. Y. Meng, "Extending Spectral Modeling Synthesis with Transient Modeling Synthesis," *Computer Music Journal*, vol. 24, no. 2, pp. 47–59, 2000.

[6] P. R. Cook, "Physically Informed Sonic Modeling (PhISM): Synthesis of Percussive Sounds," *Computer Music Journal*, vol. 21, no. 3, pp. 38–49, 1997.

[7] S. Bilbao, "A Modular Percussion Synthesis Environment," in *Proc. of the 12th Int. Conf. on Digital Audio Effects*, 2009.

[8] J. Smith and X. Serra, "PARSHL: An Analysis/Synthesis Program for non-Harmonic Sounds Based on a Sinusoidal Representation," in *Proc. of the 1987 Int. Computer Music Conf.*, 1987.

[9] T. Kirby and M. Sandler, "The evolution of drum modes with strike intensity: Analysis and synthesis

using the discrete cosine transform," *The Journal of the Acoustical Society of America*, vol. 150, no. 1, pp. 202–214, 2021.

[10] A. Ramires, P. Chandna, X. Favory, E. Gómez, and X. Serra, "Neural Percussive Synthesis Parameterised by High-Level Timbral Features," in *ICASSP 2020 - 2020 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2020.

[11] J. Nistal, S. Lattner, and G. Richard, "DrumGAN: Synthesis of Drum Sounds With Timbral Feature Conditioning Using Generative Adversarial Networks," in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.*, 2020.

[12] S. Rouard and G. Hadjeres, "CRASH: Raw Audio Score-based Generative Modeling for Controllable High-resolution Drum Sound Synthesis," in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*, 2021.

[13] F. Caspe, A. McPherson, and M. Sandler, "DDX7: Differentiable FM Synthesis of Musical Instrument Sounds," in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.*, 2022.

[14] J. Turian and M. Henry, "I'm Sorry for Your Loss: Spectrally-Based Audio Distances Are Bad at Pitch," in *"I Can't Believe It's Not Better!" NeurIPS Workshop*, 2020.

[15] B. Hayes, C. Saitis, and G. Fazekas, "Sinusoidal Frequency Estimation by Gradient Descent," in *ICASSP 2023 - 2023 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2023.

[16] R. Diaz, B. Hayes, C. Saitis, G. Fazekas, and M. Sandler, "Rigid-body sound synthesis with differentiable modal resonators," in *ICASSP 2023 - 2023 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2023.

[17] H. D. Thornburg and R. J. Leistikow, "Analysis and resynthesis of quasi-harmonic sounds: An iterative filterbank approach," in *Proc. of the 6th Int. Conf. on Digital Audio Effects*, 2003.

[18] S. N. Levine and J. O. Smith III, "A sines+ transients+ noise audio representation for data compression and time/pitch scale modifications," in *Audio Engineering Society Convention 105*, 1998.

[19] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast High-Fidelity Speech Synthesis," in *Proc. of the 35th Int. Conf. on Machine Learning*, 2018.

[20] C. J. Steinmetz and J. D. Reiss, "Efficient neural networks for real-time modeling of analog dynamic range compression," in *Audio Engineering Society Convention 152*, 2022.

[21] X. Wang, S. Takaki, and J. Yamagishi, "Neural Source-filter-based Waveform Model for Statistical Parametric Speech Synthesis," in *ICASSP 2019 - 2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2019.

[22] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An End-to-End Neural Audio Codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2022.

[23] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual Reasoning with a General Conditioning Layer," in *Proc. of the 32nd AAAI Conf. on Artificial Intelligence*, 2018.

[24] 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 - 2020 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2020.

[25] S. Birnbaum, V. Kuleshov, Z. Enam, P. W. W. Koh, and S. Ermon, "Temporal FiLM: Capturing long-range sequence dependencies with feature-wise modulations.," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[26] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[27] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.