The Drum Machine of Tao

Xiaowan Yi
Centre for Digital Music
London, UK
x.yi@qmul.ac.uk

Mathieu Barthet
Centre for Digital Music
London, UK
Aix-Marseille Univ CNRS PRISM
Marseille, France
m.barthet@qmul.ac.uk

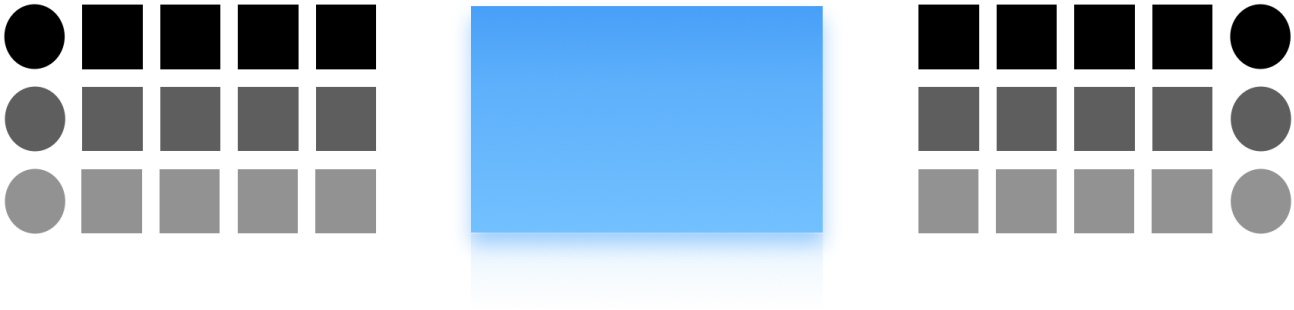

Figure 1: An abstract, symmetrical representation of a drum machine capable of generating audio waveforms from sequencer parameters and inferring sequencer parameters from audio waveforms.

Abstract

The Drum Machine of Tao (*Tao*) is a machine learning–based system that reverse-engineers sequencer parameters and one-shot percussive samples from drum loops, restoring low-level editability to sampled loops that would otherwise be frozen in audio waveforms. The philosophy behind this system is inspired from Taoism: that which returns to its primal state is the great Way of Tao. In this paper, we present the system design of *Tao*, which includes a state-of-the-art drum source separation model, a sequencer parameter estimation model, and a bespoke one-shot sample extraction algorithm that leverages differentiable audio synthesis. Results from a prototype are available for listening[1].

Keywords

drum machine, drum loop representation, sequencer parameter estimation, one-shot sample extraction

1 Introduction

Sample-based drum machines and sequencers (e.g. Akai MPC3000 [7], Elektron Digitakt[5], etc.) have been instrumental to electronic and hip-hop genres music production for decades. These systems allow musicians to craft intricate rhythms by programming multiple tracks of sequences (e.g. three tracks of sequences for kick, snare and hihats respectively), and triggering the corresponding one-shot percussive samples in time with the running sequencer, providing accessible and creative music control. However, once a drum loop is bounced or sampled into an audio waveform file, the inherent flexibility for further low-level manipulation, such as hot-swapping the one-shot kick sample or changing the sequencer patterns while keeping the same one-shot samples, is often lost. Musicians may find themselves constrained by the limitations of static, frozen loops, which lack the fine-grained editability required for more nuanced or evolving rhythm structures. On another note, the concept of making new developments on static music samples has always been important in electronic music, as artists seek to transform pre-existing material into something uniquely their own. In some cases, the use of unmodified samples is viewed as uncreative or unoriginal. Pioneering artists have mentioned that the sense of ownership of a music sample only occurs when it is deconstructed and reconfigured into something new [10]. The limitations of static, pre-sampled drum loops could impede the controllability required to personalize the original source material.

We propose the Drum Machine of Tao (*Tao*) to address this challenge by adopting machine learning techniques and reverse engineering the original representation (i.e. multitrack sequencer parameters and one-shot percussive waveforms) of a sampled drum loop. Sequencer parameters for a single track (e.g. the kick track) of 8 steps could, in its simplest form, be represented as a one-dimensional list of binary numbers (e.g. [1, 0, 0, 0, 1, 0, 0, 0]), where each number indicates the triggering state of the one-shot sample at each step. We refer to these lists as sequence activation steps, and together with tempo, they form the parameter space of a basic sequencer. Sequencer parameters and one-shot samples can also be viewed as disentangled representations of a drum loop - the former contains rhythmic information and the latter captures timbral characteristics. Drawing inspiration from Taoist philosophy, which emphasizes on returning to a primal state as a path toward fullness, *Tao* restores the low-level music interaction with sampled drum loops by estimating and extracting their elemental programmable sequences and percussive components. In the following sections, we discuss related works on drum loops information retrieval, followed by presenting our system design, implementation, and reflections with proposed future works.

[1] https://red-x-silver.github.io/the-drum-machine-of-tao/

NIME '25, June 24–27, 2025, Canberra, Australia

2 Related work

Research areas relevant to drum loop sequencer parameter estimation include Drum Transcription of Drum-only recordings (DTD) and Beat Detection (BD), both of which aim to extract rhythmic information from drums-only recordings. DTD focuses on transcribing a drum-only recording into a sequence of timestamps marking when each percussive component is struck. Its primary goal is to extract onsets and classify them into specific percussive component categories [6, 14, 16]. Beat Detection (BD), on the other hand, aims to detect beats, the evenly spaced basic rhythmic units, within music recordings. The difference between DTD and BD is that BD focuses on detecting beats, which are not always onsets, while DTD detects onsets, which do not necessarily coincide with the beats [4].

Works related to one-shot percussive sample extraction include Drum Source Separation, a sub-category of music source separation that takes a drums-only recording as input and outputs separated stems for each predefined percussive component. State-of-the-art drum source separation systems include Demucs [11] and LarsNet [8], both of which utilize deep learning techniques and provide publicly available pre-trained models.

Other non-generative applications involving drum loops, such as loop compatibility modeling, typically derive latent codes learned through neural networks as a compact representation for downstream tasks [2, 3, 15]. These compact representations of drum samples are computationally efficient, but they offer limited interpretability.

There is existing software that offers drum loop slicing functionality, including ReCycle[13] and Regroover[12]. ReCycle applies transient detection directly to sampled drum mixes and creates slices that may contain overlapping percussive elements. Regroover (now discontinued) applies source separation to the mix and enables user-controlled slicing. Their source separation algorithm produces different "layers," but without assigning specific percussive roles to each layer.

3 System design

3.1 Overview

During inference, *Tao* consists of four key components: a drum source separation model, a sequencer parameter estimation model, a differentiable rendering module, and a one-shot sample extraction module. As shown in Fig. 2, processing begins with the first two models in parallel, followed by one-shot sample extraction using the rendering module. See below for details on each component.

- A drum source separation model that demixes the drum loop into single percussive stems (e.g. kick stem, snare stem, hihats stem, etc.).
- A sequencer parameter estimation model that takes the drum loop mix as input and predicts the tempo as well as the onset times for each percussive track. The estimated onset times are then quantized into step vectors, according to the estimated tempo.
- A differentiable rendering module that takes tempo, step vectors, and one-shot percussive samples as input, and synthesizes percussive stems in audio waveform using differentiable $1D$ convolution.
- A one-shot sample extraction algorithm that operates per stem by first listing candidate one-shot percussive samples based on the estimated activation steps and the separated stem, then reconstructing the stem via the differentiable rendering module, and finally selecting the one that best reconstructs the separated stem according to a given criterion.

The final output of *Tao* consists of an estimated tempo, an estimated step vector and extracted one-shot sample for each percussive track. The proposed interface of *Tao* would be similar to any sample-based drum machine with a sequencer (e.g. a web-based TR-808[2] as shown in Fig. 3) — the estimated tempo, activation steps and extracted one-shot samples can be loaded back into the drum machine, and the sequencer is ready to run. This allows users to gain low-level editability, enabling them to transform the input drum loop (e.g. changing the rhythmic patterns while retaining the original one-shot samples, hot-swapping the one-shot samples, etc.) with full accessibility.

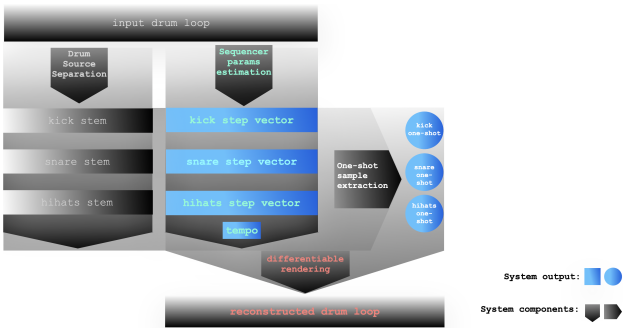

Figure 2: System diagram of *Tao*, depicting the pipeline with four key components. Note that during inference, the output from the differentiable rendering module is not the final system output; it is used solely to inform the one-shot sample extraction process.

3.2 Implementation

Our current *Tao* implementation is python-based and uses libraries including PyTorch[3]and Madmom[4].

3.2.1 Drum source separation. For the drum source separation model, we adopt the publicly available Drumsep model trained by Jarredou and Aufr33[5]. It is a Demucs model [11] trained on a customized drums dataset curated by Jarredou[6]. The Drumsep model takes two-channel audio at a 44100 Hz sample rate as input and outputs 6 audio stems of the same length as the input. We include the *Resample* API from *Torchaudio*[7] as a pre-processing step for input drum loops with sample rates different from 44100 Hz. As for mono input, we convert it into a two-channel signal by duplicating the single channel. The 6 output stems correspond to kick, snare, tom, hihats, ride, and crash respectively. We implement a simple post-processing step where we sum the hihats, ride and crash stems into a single stem that represents the broader range of cymbals.

[2]https://roland50.studio/
[3]https://pytorch.org/
[4]https://github.com/CPJKU/madmom
[5]https://github.com/jarredou/models/releases/tag/aufr33-jarredou_MDX23C_DrumSep_model_v0.1
[6]https://rigaudio.fr/datasets/DrumsDatasetv2.zip
[7]https://pytorch.org/audio/main/generated/torchaudio.transforms.Resample.html

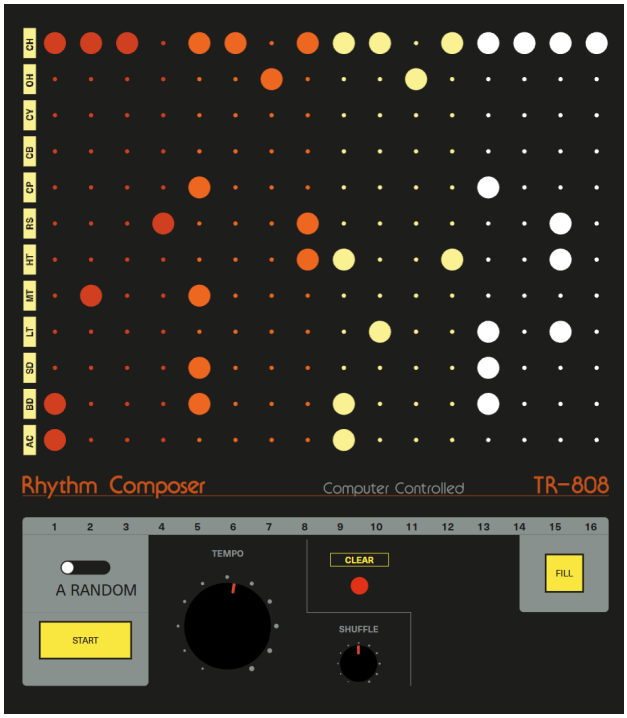

Figure 3: A screenshot of the interface of a web-based TR-808 drum machine, showing sequencer parameters such as tempo and step vectors, but excluding one-shot percussive samples.

3.2.2 Sequencer parameter estimation. Our experiment begins with a standard 8-step sequencer featuring three percussive tracks: kick, snare, and hi-hats, a common configuration in electronic music production. The global tempo range is [60, 200].

Model. We adopt a CRNN architecture similar to that of the ADTOF [16] drum transcription model consisting of a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) with Gated Recurrent Unit (GRU) layers. The input to the CNN is a spectrogram of a drum loop mix, which has 512 samples per frame and 100 frames per second. The frequency bins are transformed to a logarithmic with 12 triangular filters per octave between 20 and 8000 Hz. The input is forwarded to 4 convolutional blocks followed by 3 bi-directional GRU layers with 60 hidden units each. A final fully connected layer maps the feature dimension into the number of tracks (i.e. 3 in our *Tao* sequencer). The frame dimension remains unchanged throughout the CRNN forward pass. The immediate output from the CRNN is the predicted level of confidence in individual track's onset presence in each frame (an onset envelop in the frame dimension), which does not tell the exact discrete onset times. A post-processing step is applied after training the model to threshold the onset envelops into onset activation vectors, for which we adopt a simple peak-picking algorithm (with Madmom implementation) used in previous works [1, 16].

Additionally, we customize the CRNN for tempo estimation by attaching an extra fully connected layer with 141 neurons to the last GRU layer's output. The 141 neurons correspond to the 141 possible integer tempo values ranging from 60 to 200, as we cast the tempo estimation into a classification task. Two loss terms are computed - we use binary cross entropy loss for the estimated onset envelops and cross entropy loss for the tempo estimation. The sum of the losses are used for optimization.

Dataset. To the best of our knowledge, no sequencer-info annotated drum loops dataset is available in the public domain. One public dataset that is relevant to our study is the Freesound One-Shot Percussive Sounds ([9]) which contains 10254 one-shot percussive samples at a 16000 Hz sample rate for kick, snare and hihats among other percussive elements. To tackle this limited availability of datasets for training a sequencer parameter estimation model, we design an original data synthesis pipeline in *Tao* utilizing the publicly available one-shot sample dataset and the abovementioned differentiable rendering module as shown in Fig. 4. For synthesizing one drum loop, we randomly sample an integer within the range [60, 200] to represent the tempo, 3 binary vectors of length 8 for activations over 8 steps, and 3 one-shot samples. A 1D convolution is then applied between the one-shot samples and the activation steps leveraging the differentiable rendering module for synthesizing the drum loop. The randomly sampled tempo and step vectors are used as training targets for each synthesized loop. All drum loops are zero-padded to a length of 64000 samples which have 4 seconds of audio content at a 16000 Hz sample rate. This length ensures that even at the slowest tempo of 60 bpm in our dataset, there will still be 4 beats of content in the drum loop.

This data synthesis pipeline allows for data synthesis on-the-fly during training thanks to the parallelizability of the 1D convolution operation in the differentiable rendering module. It can also theoretically create infinitely many annotated drum loops as a means of data augmentation.

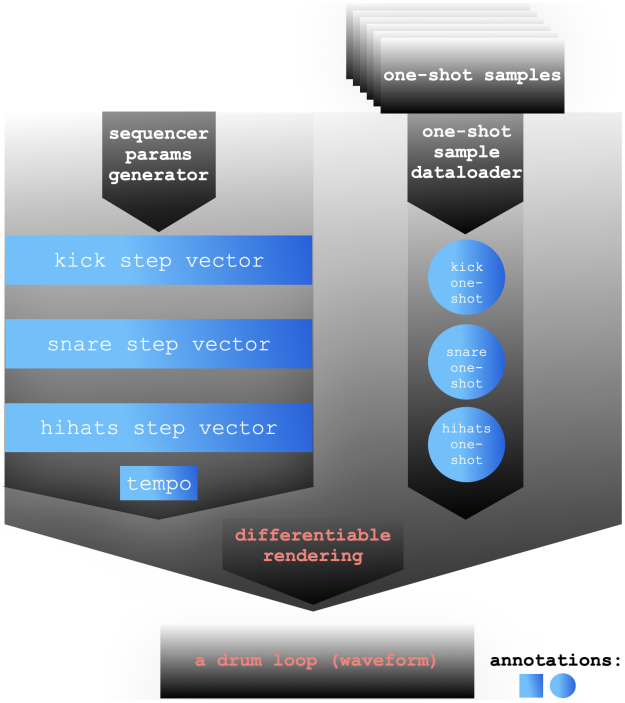

Figure 4: A diagram of *Tao*'s training data synthesis pipeline.

Training. We train the CRNN on a NVIDIA 4080 Super GPU for 50000 steps with a batch size of 64 and a learning rate of 0.0005, using ADAM optimizer. We use the abovementioned data

synthesis pipeline for synthesizing a new data batch on-the-fly at each training step.

3.2.3 One-shot sample extraction. Following the output (estimated tempo and 3-track activation steps) from the previous component, we convert the activated steps into a list of K indices in the sample space:

$[onset_0, onset_1, ..., onset_i, ..., onset_K]$. We form a candidate set of one-shot samples by slicing the source-separated stem with $[onset_i : onset_i + N]$ where N is a pre-defined length for one-shot samples. We use $N = 16000$ in our preliminary implementation. The candidate one-shot samples are then convoluted with the activation step vector to reconstruct a stem, utilizing the differentiable rendering module. We then compute the cosine similarity between the reconstructed stem and the source-separated stem in the MFCC audio feature space. Based on the similarity ranking, we select the one-shot sample that most closely reconstructs the source-separated stem.

4 Results and Future Works

Tao is a prototype system that reverse-engineers tempo, per-stem step vectors, and one-shot percussive samples from drum loops. Evaluation results on the synthesized testing set for each key component, along with audio samples and estimated sequencer parameters, are available online[8]. We are finalizing the interface and preparing to release the code[9]. We acknowledge that drum machine sequencers typically involve more nuanced parameters than binary activation vectors, including velocity, swing, and audio effects modulation. Following the initial release, we plan to extend the system to estimate parameters such as per-step velocities and swing amount.

5 Ethical Standards

This paper does not involve experiments with human or animal participants. Datasets used for training machine learning models in this paper are acquired from open access datasets.

Acknowledgments

This work is funded by UK Research and Innovation [grant number EP/S022694/1] as part of the "UKRI Centre for Doctoral Training in Artificial Intelligence and Music".

References

[1] Sebastian Böck, Florian Krebs, and Markus Schedl. 2012. Evaluating the Online Capabilities of Onset Detection Methods. In *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012*. FEUP Edições, 49–54. http://ismir2012.ismir.net/event/papers/049-ismir-2012.pdf

[2] Antoine Caillon and Philippe Esling. 2021. RAVE: A variational autoencoder for fast and high-quality neural audio synthesis. *CoRR* abs/2111.05011 (2021). arXiv:2111.05011 https://arxiv.org/abs/2111.05011

[3] Bo-Yu Chen, Jordan B. L. Smith, and Yi-Hsuan Yang. 2020. Neural Loop Combiner: Neural Network Models for Assessing the Compatibility of Loops. In *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*. 424–431. http://archives.ismir.net/ismir2020/paper/000225.pdf

[4] Matthew E. P. Davies, Sebastian Böck, and Magdalena Fuentes. 2021. *Tempo, Beat and Downbeat Estimation.* https://tempobeatdownbeat.github.io/tutorial/intro.html

[5] Elektron. 2023. *Elektron Digitakt Manual.* Retrieved February 5, 2025 from https://elektron.se/wp-content/uploads/2024/09/Digitakt_User_Manual_ENG_OS1.51_231108.pdf

[6] Olivier Gillet and Gaël Richard. 2004. Automatic transcription of drum loops. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004, Montreal, Quebec, Canada, May 17-21, 2004*. IEEE, 269–272. https://doi.org/10.1109/ICASSP.2004.1326815

[7] Roger Linn. 1994. *Akai MPC3000 Manual.* Retrieved February 5, 2025 from https://www.platinumaudiolab.com/free_stuff/manuals/Akai/akai_mpc3000_manual.pdf

[8] Alessandro Ilic Mezza, Riccardo Giampiccolo, Alberto Bernardini, and Augusto Sarti. 2024. Toward deep drum source separation. *Pattern Recognition Letters* 183 (2024), 86–91. https://doi.org/10.1016/J.PATREC.2024.04.026

[9] António Ramires, Pritish Chandna, Xavier Favory, Emilia Gómez, and Xavier Serra. 2020. Neural Percussive Synthesis Parameterised by High-Level Timbral Features. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, 786–790. https://doi.org/10.1109/ICASSP40776.2020.9053128

[10] Robert Ratcliffe. 2014. A proposed typology of sampled material within electronic dance music. *Dancecult: Journal of Electronic Dance Music Culture* 6, 1 (2014), 97–122.

[11] Simon Rouard, Francisco Massa, and Alexandre Défossez. 2023. Hybrid Transformers for Music Source Separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*. IEEE, 1–5. https://doi.org/10.1109/ICASSP49357.2023.10096956

[12] Simon Sherbourne. 2017. *Accusonus Regroover Pro.* Retrieved April 26, 2025 from https://www.soundonsound.com/reviews/accusonus-regroover-pro

[13] Reason Studio. 2025. *RECYCLE.* Retrieved April 26, 2025 from https://www.reasonstudios.com/recycle

[14] Chih-Wei Wu, Christian Dittmar, Carl Southall, Richard Vogl, Gerhard Widmer, Jason Hockman, Meinard Müller, and Alexander Lerch. 2018. A Review of Automatic Drum Transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26, 9 (2018), 1457–1483. https://doi.org/10.1109/TASLP.2018.2830113

[15] Xiaowan Yi and Mathieu Barthet. 2024. A Generative Framework for Composition-aware Loop Recommendation In Music Production: Drum2Bass Use Case. In *Proceedings of the 21st Sound and Music Computing Conference. July 4-6, 2024. Porto, Portugal.*

[16] Mickaël Zehren, Marco Alunno, and Paolo Bientinesi. 2021. ADTOF: A large dataset of non-synthetic music for automatic drum transcription. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, Jin Ha Lee, Alexander Lerch, Zhiyao Duan, Juhan Nam, Preeti Rao, Peter van Kranenburg, and Ajay Srinivasamurthy (Eds.). 818–824. https://archives.ismir.net/ismir2021/paper/000102.pdf

[8] https://red-x-silver.github.io/the-drum-machine-of-tao/

[9] https://github.com/red-x-silver/tao