Audio Engineering Society

Late Breaking Demo Paper

Presented at the AES International Conference on
Artificial Intelligence and Machine Learning for Audio
2025 September 8–10, London, UK

Latent rhythm transformation of drum recordings

Jason Hockman[1] and Jake Drysdale[2]

[1] *School of Digital Arts, Manchester Metropolitan University, Manchester, United Kingdom*
[2] *Beathoven.ai, Bengaluru, Karnataka, India*

Correspondence should be addressed to Jason Hockman (`j.hockman@mmu.ac.uk`)

ABSTRACT

A method is proposed for rhythm style transfer of multitimbral drum recordings via conditioning a VAE on rhythm and timbral features. Modulation and estimation of latent parameters and a novel resequencing process for reconstruction loss result in an end-to-end transformation circumventing manual segmentation and alignment.

1 Introduction

Drums play a crucial role in shaping the rhythmic and timbral identity of many forms of music. In multitimbral drum recordings, overlapping events, expressive timing, and timbral subtlety make rhythmic structure difficult to isolate and manipulate. Many professional studio workflows rely on *redrumming*, a technique that replaces or layers recorded drums with alternative recordings while preserving the original timing of a target performance [1]. Dedicated programs such as Recycle and modern DAWs (Logic, Ableton) provide manual or semi-automated workflows for time-based slicing and manipulation of waveforms; however, success of such processes is limited by spectral overlap of drums, requiring time-consuming manual intervention. In this paper, we extend the well-known RAVE method [2] to transformation of the rhythmic characteristics of a source drum recording to match the timing and drum classes present in a target recording.

2 Method

An overview of the proposed system is presented in Figure 1. The system operates in two stages: (1) VAE training and (2) transformer rhythm resequencing. For Stage 1, we adopt the RAVE [2] encoder E and generator G, training them on 3-sec drum recordings x_s to produce a latent representation z from the encoder outputs—mean μ and standard deviation σ. Stage 2 discards σ and introduces a lightweight transformer stack that performs style transfer on μ, yielding $\hat{\mu}$ and a subsequent estimation of $\hat{\sigma}$. Attention key k and value v are learned projections of μ, and query q is derived from conditional features c_t obtained from 5-class drum transcription (ADT) probabilities [3], $r \in R^{B\times 5\times T}$, and intermediate encoder activations $a \in R^{B\times C\times T}$ from target drum recordings x_t. These are projected into embeddings r_e and a_e via Conv1D sub-networks with LeakyReLU activations, to provide non-prescriptive attention guidance. Both r_e and a_e use four-layer Conv1D stacks

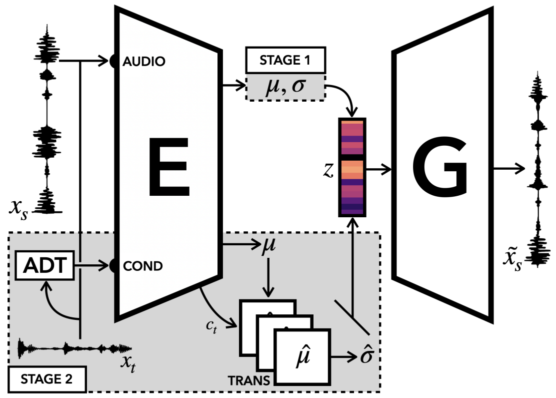

Fig. 1: Proposed model for rhythm transformation.

with LeakyReLU, with r_e kernel size 5, a_e size 3, each mapped to $2D$ channels ($D = 128$). Outputs are merged via gated fusion and projected to form q:

$$\lambda = \phi(W_g[r_e; a_e]), \qquad q = \lambda r_e + (1-\lambda)a_e. \tag{1}$$

$[\cdot;\cdot]$ denotes channel-wise concatenation, W_g is a learnable linear layer, and ϕ is the sigmoid function. The resulting query q is passed to a stack of L=3 transformer blocks with $H = 4$ attention heads each. Layerwise cross-attention is applied using projections of the current $\mu^{(l)}$ in k and v:

$$\mathrm{Attn}(q,k,v) = \mathrm{softmax}\left(\frac{qk^\top}{\sqrt{d}} + \gamma B_{rel} + \delta B_{rhythm}\right) v, \tag{2}$$

where $\gamma, \delta \in R^H$ are learned per-head scaling factors, $B_{rel} \in R^{B\times H\times T\times T}$ is a learned relative positional bias, $B_{rhythm} \in R^{B\times H\times T\times T}$ is a rhythm bias derived from a linear projection of ADT probabilities and d is per-head dimensionality ($d = \frac{D}{H}$). $\hat{\mu}$ is updated at each layer l via linear interpolation using a learned scalar coefficient $\alpha^{(l)} \in [0,1]$. After the final layer, $\hat{\sigma}$ is estimated from $\hat{\mu}$ via convolutional projection f_σ:

$$\hat{\sigma} = \mathrm{softplus}\left(f_\sigma * \hat{\mu}^{(L)}\right) + \varepsilon, \tag{3}$$

where $*$ is 1D convolution and f_σ consists of two 1×1 Conv1D layers with LeakyReLU activation. The resulting $\hat{\mu}$ and $\hat{\sigma}$ parameterise a Gaussian posterior from which z is sampled and passed to G yielding $\tilde{x}_s$.

3 Training

Stage 1 follows the representation learning and adversarial training procedure in [2] with 20,000 3-sec segments of real and synthesized drum recordings. Stage 2 guides rhythm transform learning while concurrently *re*-regulating the latent space over 6000 steps. The model is trained using Adam ($LR = 2e^{-4}$) with batch size 8; fuser networks are trained with a reduced rate of $1e^{-4}$ to mitigate timbre leakage. As the transform modifies source rhythmic–timbral layout, standard reconstruction loss (i.e., comparing $\tilde{x}_s$ to x_s) is unsuitable. To produce pseudo-targets for training reconstruction losses, Stage 2 employs a 10,000-segment synthetic dataset created from 100 kits constructed from real and electronic oneshots with pitch and gain augmentation and 100 rhythms across various styles. Temporal expressivity is added via tempo scaling, swing, microtiming, and event-timing dropouts. Kits are stored per-segment with pitch and gain settings, enabling resequenced reconstruction loss pseudo-targets by applying augmented oneshots used in assembling x_s to drum class events and timing determined by x_t. Cycle consistency and attention entropy losses respectively promote invertibility and head diversity during training. Following Stage 2, the trained system performs transformations directly on real drum recordings.

4 Examples and Summary

Examples of the transformation are presented on the supporting website.[1] System output sounds coherent, with the rhythm of the target applied as intended to the source timbres. In cases where the timbre is not convincingly achieved in the generations, traces of the source are heard with reduced high frequency detail; however, improved fidelity is expected through further experimentation with loss balancing. Future work will explore an interactive sequencing interface to allow direct control over target rhythm events within x_s.

References

[1] López-Serrano, P., Davies, M. E., Hockman, J., Dittmar, C., and Muller, M., "Break-informed audio decomposition for interactive redrumming," in *ISMIR*, 2018.

[2] Caillon, A. and Esling, P., "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," arXiv:2111.05011, 2021.

[3] Zehren, M., Alunno, M., and Bientinesi, P., "High-Quality and Reproducible Automatic Drum Transcription from Crowdsourced Data," *Signals*, 4(4), 2023.

[1] `https://jhockman.github.io/`