FROM ARBITRARY AUDIO TO EDM: AUDIO-CONDITIONED RETRIEVAL OF DISCRETE RHYTHM ARCHETYPES

Lindsey Pietrewicz

New York University
New York, USA
lgp3212@nyu.edu

ABSTRACT

We present a system for transforming arbitrary audio into Electronic Dance Music (EDM) drum patterns while preserving the timbral identity of the source material. A Vector Quantized Variational Autoencoder (VQ-VAE) trained on 7,999 EDM drum loops learns a discrete codebook of 256 rhythm archetypes, validated through UMAP and hierarchical clustering to exhibit semantically meaningful structure. At inference, spectral features extracted from arbitrary input audio select the nearest archetype via nearest-neighbor retrieval in a shared audio feature space. A training sample from the selected archetype is reconstructed through the VQ-VAE, and a second decoder predicts per-hit velocity dynamics. The user's sounds are then placed at the reconstructed hit positions, scaled by predicted velocity. Applied to 2,000 files from the ESC-50 environmental sound dataset, the system activates 128 of 256 codebook entries (50% coverage), demonstrating broad responsiveness to diverse non-EDM audio. Audio examples are available at: https://lgp3212.github.io/edm-style-transfer/.

1. INTRODUCTION

Electronic Dance Music achieves groove not merely through the precise temporal placement of events on a quantized grid, but through the continuous sonic parameters that shape rhythmic perception. Brøvig-Hanssen et al. demonstrated that EDM producers working with strict temporal quantization create rhythmic variation primarily through sound design—amplitude envelopes, attack shaping, and volume automation—rather than microtiming deviations [1]. Complementing this, Danielsen et al. established that perceived timing depends fundamentally on acoustic characteristics: attack time, duration, and spectral content all influence when listeners perceive an event to occur [2]. These findings suggest that a computational model of EDM rhythm must capture both temporal structure and dynamic characteristics.

Prior work on rhythmic pattern learning has demonstrated the effectiveness of discrete representations. Han et al. applied VQ-VAE to symbolic drum loop generation, achieving superior results over GANs and transformers by compressing MIDI patterns into discrete latent codes [3]. However, their work operates entirely in the symbolic domain and focuses on generation from scratch rather than transformation of existing audio material.

We address the distinct problem of *cross-domain rhythm transfer*: given arbitrary input audio, produce an EDM drum pattern that reflects the spectral character of the input while placing the user's own sounds at EDM-appropriate positions. The system makes three contributions: (1) a learned discrete space of EDM rhythm archetypes with demonstrable semantic structure, (2) audio-conditioned retrieval in a shared spectral feature space, and (3) a velocity decoder that captures the dynamic character of EDM percussion identified as perceptually crucial by Brøvig-Hanssen et al.

2. SYSTEM DESCRIPTION

The pipeline consists of four stages: VQ-VAE codebook learning, audio conditioning and retrieval, velocity decoding, and audio synthesis. Figure 1 illustrates the learned codebook structure.

2.1. Phase 1: VQ-VAE Rhythm Codebook

The VQ-VAE [4] is trained on binary drum patterns from the EDM-HSE dataset: 7,999 samples represented as (27×64) pianorolls covering 27 drum classes at 16th-note resolution over 4 bars. Patterns are binarized prior to training.

The encoder consists of two strided CNN blocks (stride 2) with residual connections, compressing the 64-step sequence to a 16-step latent representation of dimension $d = 16$. The quantizer maps each latent vector to its nearest entry in a codebook $\mathcal{C} = \{e_k\}_{k=1}^{256}$ using Exponential Moving Average updates [4] with random restart at threshold $\tau = 0.5$ to prevent codebook collapse. The decoder mirrors the encoder with upsampling layers. Training uses:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \beta \|\text{sg}[z_e] - e\|_2^2, \quad \beta = 0.25 \tag{1}$$

Training for 300 epochs on an NVIDIA A10G GPU yields best validation Hamming distance of 0.0064 at epoch 94. Codebook utilization reaches 100% with perplexity $84.3/256$, confirming no collapse.

2.2. Codebook Analysis

To validate that the codebook captures semantically meaningful structure, we apply UMAP [5] and hierarchical clustering to the 256 codebook vectors. Figure 1 shows eight well-separated clusters corresponding to musically distinct archetypes: kick-dominant grooves (C0), snare-heavy fills (C1), maracas/textural patterns (C2), hi-hat closed dense (C3), clap-driven backbeats (C4), open hi-hat sparse (C5), hi-hat closed mid (C6), and kick groove (C7). Usage follows a power-law distribution, with a small number of entries activated frequently and a long tail of specialized entries. This semantic organization motivates the use of nearest-neighbor retrieval for audio conditioning.

2.3. Phase 2: Velocity Decoder

Following Brøvig-Hanssen et al.'s finding that EDM dynamics are perceptually central [1], we train a second decoder to predict per-hit velocity from continuous velocity data ($v \in [0, 1]$), with the Phase 1 encoder and quantizer frozen. Loss is computed only at hit positions:

$$\mathcal{L}_v = \frac{\sum_{i,t} \mathbf{1}[x_{i,t} > 0] \cdot (\hat{v}_{i,t} - v_{i,t})^2}{\sum_{i,t} \mathbf{1}[x_{i,t} > 0]} \tag{2}$$

The decoder achieves MAE $= 0.0845$ on held-out data, a 77.4% reduction over a mean-prediction baseline (MAE $= 0.374$). The training velocity distribution is bimodal, with peaks at $v < 0.2$ (ghost notes) and $v > 0.8$ (hard accents), consistent with EDM production practice. The decoder captures the predominant hard-hit character of EDM (65% of predicted velocities above 0.8), though underrepresentation of ghost notes remains a limitation we discuss in Section 5.

2.4. Audio Conditioning and Retrieval

To bridge the gap between arbitrary input audio and the EDM codebook, we precompute spectral signatures for each of the 256 codebook entries in audio feature space. Specifically, for each of the 7,999 training samples we extract four librosa [6] features: spectral centroid (Hz), RMS energy, spectral rolloff (Hz), and zero-crossing rate. Features are averaged over all samples whose dominant codebook index matches each entry, yielding a (256×4) signature matrix in true audio space. This lightweight feature representation was chosen to support real-time interaction during live demonstration.

At inference, the same four features are extracted from onset segments of the input audio and averaged into a global feature vector. Nearest-neighbor retrieval in the normalized feature space selects the starting codebook entry:

$$k^* = \arg\min_k \left\| \tilde{f}_{\text{audio}} - \tilde{s}_k \right\|_2 \tag{3}$$

where $\tilde{f}_{\text{audio}}$ and $\tilde{s}_k$ are normalized audio and signature vectors respectively.

2.5. Audio Synthesis

A training sample assigned to codebook entry k^* is passed through the full VQ-VAE forward pass (encoder → quantizer → decoder), producing a dense binary rhythm grid. The velocity decoder then predicts dynamics at each hit position. Input audio onset segments are clustered into 9 groups via k-means on spectral features, with each cluster mapped to a drum voice. The number of clusters k=9 is chosen as a practical default providing sufficient timbral variety to cover the main EDM drum voice categories (kick, snare, hi-hats, clap, toms) without over-segmenting short input recordings. User audio segments are placed at hit positions scaled by predicted velocity, with reverb applied to the output.

3. EVALUATION

Codebook generalization. To evaluate whether the learned codebook generalizes beyond EDM training data, we applied the audio conditioning pipeline to all 2,000 files from the ESC-50 environmental sound dataset [7], spanning 50 categories across five super-categories (Animals, Nature, Human, Domestic, Urban). Figure 2 shows the activation results. The system activated 128 of 256 codebook entries (50% coverage), with the most frequently activated entries showing dominance by specific acoustic categories: impulsive domestic sounds (door knocks, glass breaking) cluster toward distinct codes from natural transients (rain, water drops) and urban machinery (sirens, engines). Nature sounds activated the most unique codes (83), while Animals activated the fewest (49), suggesting that percussive non-EDM sounds navigate the rhythm archetype space more broadly than animal vocalizations.

Table 1: Phase 1 VQ-VAE reconstruction results on held-out validation data.

Metric	Value
Best validation Hamming distance	0.0064
Codebook utilization	100%
Codebook perplexity	84.3/256
Velocity decoder MAE	0.0845
Velocity baseline MAE	0.374
Improvement over baseline	77.4%

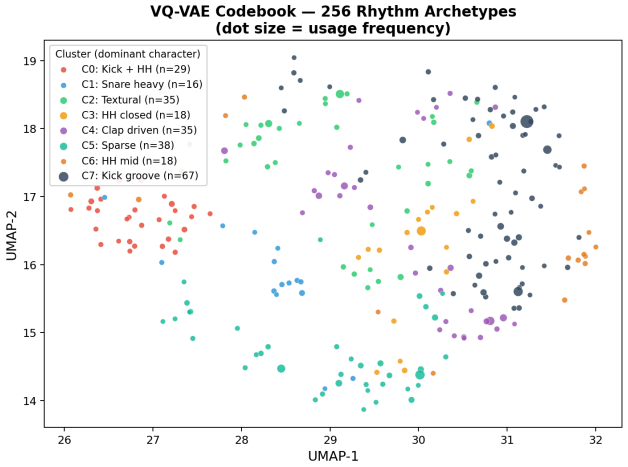

Figure 1: UMAP projection of 256 VQ-VAE codebook vectors colored by cluster. Dot size encodes usage frequency. Eight semantically distinct rhythm archetypes emerge with clear spatial separation, confirming the codebook has learned musically meaningful structure.

4. DEMONSTRATION

The demonstration presents the system running live on a laptop. Visitors provide arbitrary audio — speech, singing, found sounds, instrument recordings — and hear an EDM drum pattern generated using their own sounds. The demonstration highlights: (1) timbral preservation — output audio uses the visitor's own sounds, not a drum kit; (2) audio conditioning — spectrally distinct inputs produce different codebook entries and thus different rhythmic

Audio Conditioning: Input Sound → Codebook Archetype

Input sound	Category	Codebook entry	Rhythm archetype
Dog bark	Animals	117	Kick + HH
Rain	Nature	205	Sparse / textural
Crackling fire	Nature	205	Sparse / textural
Siren	Urban	232	HH closed (mid)
Helicopter	Urban	202	Kick + HH
Keyboard	Domestic	48	Clap driven
Footsteps	Human	250	Sparse
Church bells	Urban	25	Kick + HH
Glass breaking	Domestic	110	Snare / impact
Chirping birds	Nature	244	HH dense

Animals Nature Human Domestic Urban

Figure 2: Ten diverse input sounds mapped to codebook archetypes via spectral nearest-neighbor retrieval. Acoustically similar sounds (i.e. rain and crackling fire) map to the same archetype, demonstrating that retrieval responds to spectral character rather than semantic label.

patterns; (3) velocity expressiveness — the output exhibits dynamic variation rather than flat-velocity playback.

Audio examples including full producer remixes built from system output are available at: `https://lgp3212.github.io/edm-style-transfer/`. Audio examples spanning speech, environmental sounds, instrumental recordings, and found sounds are provided on the accompanying website. These examples provide qualitative evidence that the retrieval mechanism maps diverse source materials to rhythm archetypes while preserving source timbre.

System requirements: a standard laptop running Python 3.11 with librosa, PyTorch, and soundfile.

5. DISCUSSION AND FUTURE WORK

The system demonstrates that a VQ-VAE codebook trained on EDM patterns serves as a semantically organized rhythm archetype index navigable by spectral features of arbitrary audio. The 50% codebook coverage on ESC-50 suggests broad generalization beyond the EDM training domain.

Limitations include the ghost note underrepresentation in the velocity decoder, attributable to the masked MSE objective's equal weighting of all hit positions regardless of dynamic magnitude. A loss function that explicitly models the bimodal velocity distribution — for example, a mixture density loss — would better capture the full dynamic range. The audio-to-codebook feature mapping is also heuristic; a learned cross-modal embedding trained on paired (audio, pattern) data would improve conditioning fidelity.

Future work includes training an autoregressive LSTM prior over codebook index sequences to enable novel pattern generation beyond retrieval, and extending the conditioning to longer-context audio features (tempo, energy trajectory) for more musically informed archetype selection.

6. ACKNOWLEDGMENTS

The author thanks Iran Roman and the Centre for Digital Music at Queen Mary University of London for research mentorship, and Beau Bialow for production assistance with the audio demonstration examples.

7. REFERENCES

[1] R. Brøvig-Hanssen, B. Sandvik, J. M. Aareskjold-Drecker, and A. Danielsen, "A grid in flux: Sound and timing in electronic dance music," *Music Theory Spectrum*, vol. 44, no. 1, pp. 1–16, 2022.

[2] A. Danielsen, R. Brøvig, K. K. Bøhler *et al.*, "There's more to timing than time: Investigating musical microrhythm across disciplines and cultures," *Music Perception*, vol. 41, no. 3, pp. 176–198, 2024.

[3] S. Han, H. Ihm, M. Lee, and W. Lim, "Symbolic music loop generation with neural discrete representations," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 403–410.

[4] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[5] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[6] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in Python," in *Proceedings of the 14th Python in Science Conference (SciPy)*, 2015.

[7] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015.