Tapping Into a New Paradigm: A Synthetic Strategy for Automatic Drum TapScription

André C. Santos
andresantos@dei.uc.pt
CISUC - University of Coimbra
Coimbra, Portugal

Amílcar Cardoso
amilcar@dei.uc.pt
CISUC - University of Coimbra
Coimbra, Portugal

Matthew E. P. Davies
Independent Researcher
Portugal

Roger B. Dannenberg
rbd@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Figure 1: Desk drumming, a form of tapping or drumming on a surface, often using the hands to create rhythmic sounds, is typically seen in informal, creative contexts.

Abstract

We introduce Automatic Drum TapScription (ADTS), a novel paradigm for rhythmic interaction consisting of transcribing arbitrarily-timbred taps into drum representations. Our approach targets taps produced on a variety of surfaces without other controlled timbral characteristics other than playing style. Our long-term goal is to enable more accessible and creative percussive exploration, but this presents significant challenges due to the minimal timbre variation between taps intended to represent different drum classes. To address these challenges, we take the first steps toward achieving ADTS by designing an effective dataset synthesis strategy. This strategy enables new opportunities for musical expression by considering drumming at a more semantic or functional level as opposed to a simple collection of timbres. We present initial results, comparing three different models: one trained on drum data, another trained on a small dataset of quasi-aligned tapped performances, and another trained on our synthetic dataset. Our synthetic approach shows promise, demonstrating progress in this untapped domain.

Keywords

Automatic Drum Transcription, Rhythmic Interaction, Synthetic Dataset, Musical Expression

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

1 Introduction

Among the many ways of interacting with music, rhythm-based interactions - such as feeling the beat by nodding, clapping, or tapping - are arguably the most instinctive [19]. This is not surprising - we are inherently rhythmic beings. Our heartbeat, breathing, and walking pace, three essential and omnipresent aspects of our lives, are all rhythmic [9, 13, 17, 18]. Among rhythmic interactions, a more complex form is the replication or improvisation of rhythmic patterns or grooves via finger tapping, desk drumming (see Figure 1) or even air drumming. Indeed, any solid surface is capable of serving as a percussive vehicle for conveying rhythmic ideas.

However, very little work has been done to study or capitalize on this common rhythmic expression form. Given the significant recent advancements in deep audio and rhythmic modelling techniques [3, 16] we consider rhythm-based interaction to be a promising area for exploration. For example, O'Reilly et al. [22] demonstrate this type of interaction by taking advantage of Descript Audio Codec (DAC)'s [16] modeling capabilities and leverage a masked token training approach to present The Rhythm in Anything (TRIA), "a system for mapping arbitrary percussive sound gestures to high-fidelity drum recordings". However, it still is a preliminary approach and as such has limitations that

may hinder user experience, namely the controllability, repeatability and editability aspect of the system. In previous work, we showed that off-the-shelf drum timbre transfer does not work well for every type of percussive timbres [25]. To make the most of this interaction, we argue for a hybrid combination of symbolic and audio approaches. Symbolic representations are much easier to represent, manipulate and control but may lack expressivity when compared to raw audio recordings. Audio representations, by contrast, capture expressive subtleties of performance but are much more challenging to manipulate while preserving audio quality.

In this work, we focus on the symbolic aspect of this interaction and propose a new task, Automatic Drum TapScription (ADTS), to lay the foundation for the study of this kind of rhythmic interaction. ADTS is similar to Automatic Drum Trancription (ADT), however it is not a direct transcription because the different input sounds (e.g., taps on a desk) are not expected to be perceptually similar to the target drum sounds (e.g., a snare drum). Furthermore, a practical motivation for initially focusing on transcription rather than direct audio-to-audio transformation is the availability of data. Audio-to-audio models typically require vast amounts of training data, and, to our knowledge, there are no existing paired datasets of arbitrarily timbred tapped rhythms and their corresponding drum patterns. So we first establish a robust data synthesis strategy to tackle ADTS which can then serve as a foundation that facilitates the future extension of our approach to audio-to-audio models. Our data synthesis strategy is used to extrapolate upon an initial dataset of just 22 minutes of tapped audio recordings.

The paper is structured as follows: in Section 2 we review ADT and other relevant work in the field of symbolic drum generation; then, in Section 3 we formally define ADTS, address some of its challenges and particularities, and outline some potential applications. In Section 4, we introduce our synthetic strategy to address the lack of data: from obtaining a collection of one-shot tap sounds to the actual synthesis process itself. We then present our results in section 5 where we explain how we trained three Temporal Convolutional Network (TCN) models on different datasets and compare their performances. Finally, in Section 6 we discuss our results, present conclusions, and identify promising future work.

2 Related Work

2.1 Automatic Drum Transcription (ADT)

Automatic Music Transcription (AMT) consists in transcribing a musical piece based on its audio. It can be thought of as reverse engineering a musical performance, i.e., instead of having a score and producing an audio output, transcription transforms an audio recording into some symbolic representation (not necessarily a traditional musical score).

ADT is then the machine learning task of transcribing drum instruments. It was first introduced at MIREX 2005 as Audio Drum Detection [30] and refined to its modern definition at MIREX 2017 [23], advancing the SOTA with more recent Deep Learning (DL) techniques.

Wu et al. [35] made a comprehensive review of ADT systems, arranging them into four categories: Segmentation-Based approaches, Classification-based approaches, Language-Model-based approaches, and finally Activation-based approaches. Here we focus only on the latter group as it showed the highest overall performance.

The rationale behind activation-based methods is to produce an activation function that indicates the likelihood of a drum instrument being present along the duration of the recording. Given this activation function, detecting events for each instrument can be performed with simple peak-picking strategies, similar to Onset Detection (OD) [1]. This approach can be further divided into two different subcategories: matrix factorisation approaches and DL. In general, Machine Learning (ML) approaches are data-centric and seldom rely on any other previous information or knowledge, other than the correct labelling of data. This also makes their usability and overall performance dependent on the training data which can lead to *overfitting* and hence poor generalization to unseen data. Nevertheless, they have proven to be reliable at performing ADT, particularly Recurrent Neural Networks (RNNs) [28, 31] and Convolutional Recurrent Neural Networks (CRNNs) [32], which are apt at modelling time series data. Other architectures like Convolutional Neural Networks (CNNs) have also been employed successfully [14, 29], as well as transformers and self-attention mechanisms [12]. Finally, TCNs have also been gaining popularity in rhythm related tasks [6] due to their efficiency and relatively high performance.

Recent developments have been made regarding unsupervised learning approaches to try to circumvent the absence of large enough labelled datasets required by supervised approaches. Wang et al. [34] used a *semi-supervised* approach called *few-shot learning* that can recognise previously unseen instrument classes with minimal input from humans. Choi and Cho [5] were able to compete with current SOTA systems using a completely unsupervised approach. Another way to cope with the lack of data is use data augmentation [20]. Rohit M A et al. [24] used transfer learning techniques on current ADT models and relied heavily on data augmentation to build a Tabla transcription system. Efforts have also been made recently by building new large datasets [4] which have been studied and data engineered to try to build more robust and reliable ADT systems [11].

2.2 Symbolic Drum Generation

Within the field of symbolic drum generation, Gillick et al. [8] pioneered a self-supervised approach to train Seq2Seq models. They proposed and tackled novel tasks such as Humanization, Infilling, and Tap2Drum. They also recorded a drum dataset, the Groove MIDI Dataset (GMD), consisting of "over 13 hours of recordings by professional drummers aligned with fine-grained timing and dynamics information". The dataset includes both audio and symbolic files with velocity information across several music genres, rhythmic patterns and drum fills. Tap2Drum in particular can be considered the symbolic equivalent of ADTS: a model is trained to infer a drum pattern and groove from a monophonic sequence of note hits. This is not trivial because of sound superposition and absence of timbral information.

The work by Gillick et al. allowed for exciting further explorations in terms of symbolic drum generation, namely in terms of generation and continuation [21], as well as infilling and variation [10]. In [21], the authors explored transformers for generating and continuing symbolic drum patterns. The results are comparable to human drummers. Then, in [10], Haki et al. provided a creative use of this approach by iteratively generating variations of a given drum pattern via infilling. This allows for creative exploration of several rhythmic patterns by users.

3 Automatic Drum TapScription (ADTS)

Inspired by [33, 35], we now define the task of ADTS, its challenges and particularities, and some application scenarios in subsections 3.1, 3.2 and 3.3, respectively.

3.1 Task Definition

Following [33], the task of Automatic Drum Trancription (ADT) is defined as follows:

> **Drum transcription** is the task of detecting the positions in time and labelling the drum class of **drum instrument onsets in polyphonic music**.

Adapting that definition, we can then define the task of ADTS as:

> **Drum TapScription** is the task of detecting the positions in time and labelling the drum class of **arbitrarily-timbered tapped onsets in percussive inputs, assuming an underlying drum pattern.**

As with early work in ADT, we consider only Kick Drum (KD), Snare Drum (SD) and Hi-Hat (HH), but more drum classes can be added in the future.

Evaluation: The primary evaluation metric is the F-measure [26], computed separately for each drum type (KD, SD, and HH) and as a total F-measure across all instrument classes. The F-measure is the harmonic mean of precision and recall, with $\beta = 1$, meaning equal importance is given to both. Onset deviations (between estimated note position and ground truth) are considered correct if they fall within a ±20 ms tolerance window.

3.2 Challenges and Particularities

Desk drumming or finger tapping is a subtle skill. For one, emulating a drumkit with only two hands is inherently limited, so conveying the idea of a groove requires some learning. Individual taps may sound similar in isolation, but when executed with musical skill, a groove emerges when considered in context. This subtlety is one of the greatest challenges of ADTS: intra-pattern timbre differences can be negligible, whereas in a real drum kit, each class is generally timbrically distinct. In other words, a regular KD sounds much more different from a regular SD than a "tap" KD from a "tap" SD. Figure 2 highlights how much clearer the frequency activations are for an original drum recording compared to its tapped performance, underscoring the difficulty of classifying hits. This challenge is inherently linked to the tapper's performance and the temporal structure of a tapping pattern. This makes incorporating temporal structure into ADTS models crucial in our opinion. That is why we use a TCN, hoping that this type of architecture is able to capture longer temporal dependencies than other architectures (e.g. CNNs).

Conversely, and somewhat paradoxically, considering the nearly limitless combinations of playable surfaces, recording conditions, user playing styles, or props used (e.g. drumsticks, pens, specific different timbred objects such as a glass), inter-pattern timbre differences may be much larger for ADTS than for ADT. In other words, two different tapping surfaces might be much more different than two different drumkits. Pairing that with accommodating both proficient tappers and beginners, generalizability becomes a serious challenge. This evokes a need for robustness in ADTS systems. We believe that considering relative timbre differences is paramount to address this challenge in ADTS.

Finally, there is no perfectly aligned, large-scale dataset of tapped performances and corresponding drum patterns. This makes it difficult to apply traditional DL strategies, even though we know this type of approach yields the best results in ADT and other music analysis tasks like Source Separation (SS) [7, 35]. One can always record an aligned dataset of tapped patterns and drum patterns, but given the meticulousness with which such a dataset needs to be created (e.g., near-perfect-time alignment, recording conditions, different playing styles), recording a dataset like this seems to be an extremely arduous task. One solution is to automatically synthesize a dataset from a smaller set of tapped recordings. In Section 4, we detail our strategy for synthesizing a paired tapped dataset to enable traditional supervised learning approaches.

3.3 Interaction Potential and Application Scenarios

While it is true that one can reliably and accurately go from tapping to MIDI drums via drum pads (an interaction known as finger drumming; not to be confused with finger tapping nor desk drumming), we argue that ADTS can complement or even surpass this form of interaction - especially when paired with further timbre transfer techniques - for the following reasons: accessibility, expressivity, creativity, and innovation in musical education.

First, by decoupling the interaction from specialized hardware such as drum pads, conveying rhythmic ideas using only a smartphone's or computer's microphone becomes accessible to almost anyone - like having a plug-and-play virtual drumkit in your pocket at all times; a "drummer's sketchbook", if you will, allowing drummers to tap out preliminary ideas for drum patterns, which can later be refined in a DAW or recorded with a traditional drum kit.

Second, by not being limited by physical buttons or touch surfaces, users can enhance their expressivity with subtleties via their playing style. Furthermore, one can also explore every kind of surface as a potential percussive vehicle, leading to experimentation in the timbre space of the surfaces used and how they are struck. This experimentation and subtleties can later be captured as well by timbre transfer audio models [3]. A method for accurate ADTS would enable more precise control of similar models.

Third, this new paradigm might elicit new creative mediums, input sounds, playing styles, recording conditions, or collaborations. For example, after achieving ADTS, it is not hard to imagine a system that can assist in rhythmic pattern variation or continuation in the symbolic domain [10, 21].

Finally, by pairing an ADTS system with musical notation, we can see the potential a tool like this can have in teaching rhythm and drumming concepts to students.

4 Dataset Synthesis

The goal of this section is to describe the synthesis of our dataset by detailing the motivation behind our approach, the methods employed to classify and extract one-shot tap sounds, and the synthesis process used to generate natural-sounding tap sequences. The purpose of the dataset creation step is to produce a large amount of synthetic tapped rhythmic patterns that can be used to train a supervised model for ADTS. A particular challenge is to produce "natural-sounding" synthetic tap sequences that are perceptually similar to real recordings of tapped rhythms.

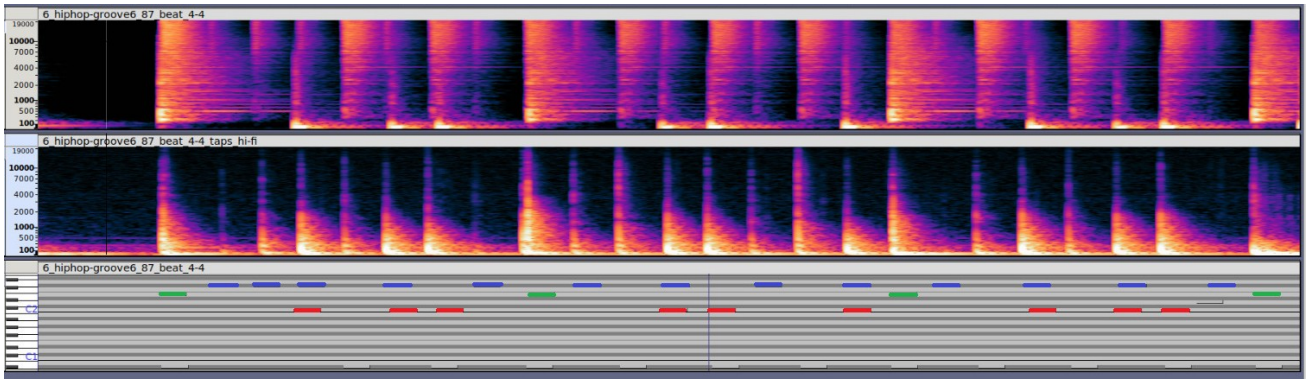

Figure 2: Intra-pattern similarity. Top row: Original drum recording from GMD; Middle row: quasi-aligned tapped recording of the same pattern; Bottom row: Corresponding MIDI file - The colour code is as follows: KD in red, SD in green, and HH in blue.

	Peak Amplitude	Loudness	Sound Pressure Level (SPL)	Spectral Centroid	Temporal Centroid	Spectral Flatness
KD	55.70	55.88	**27.41**	1189.70	55.84	55.92
SD	67.73	67.89	**16.14**	1201.75	67.87	67.94
HH	67.41	67.55	**19.07**	1150.99	67.53	67.59

Table 1: Wasserstein-1 distances (EMD) between predicted and ground-truth distributions for each normalized audio feature across three drum classes: kick drum (KD), snare drum (SD), and hi-hat (HH). Since all features and velocity values were normalized prior to distance computation, the reported EMD values are unitless. SPL consistently achieves the lowest EMD, indicating it is the most effective feature to map to the MIDI notes' velocity.

4.1 Core Idea

The GMD [8] provides not only raw drum recordings but also corresponding MIDI files. Our objective is to synthesize these MIDI files using appropriate - i.e. with the correct drum class - one-shot tap sounds, creating a dataset of corresponding tap sequences. However, a fundamental challenge arises: how to classify tap sounds accurately when the very model we aim to train requires labeled taps to begin with — creating a circular problem.

4.2 Obtaining Labeled Tap One-Shots

We explored two different approaches to obtain labeled one-shot samples: One method involved recording a continuous stream of tap performances, then manually listening to each tap and labeling it as KD, SD, or HH. However, this approach proved to be time-consuming, highly subjective, and prone to errors, making it impractical for large-scale dataset creation. A more efficient approach involved the first author (an experienced drummer) recording a small set of tapped performances while listening to the corresponding audio drum tracks from the GMD dataset. This method of learning and playing by ear is common among drummers. By doing so, we obtained a paired dataset of 22 minutes with each tap sound aligned with its corresponding labelled MIDI event.

To expand this dataset further, we extracted individual one-shot samples from the recordings. Since each tap was performed while listening to a drum track, the classification of each one-shot was inherently known. However, real-world recordings introduce imperfections, such as timing mismatches and variability in performance. To refine our dataset, we employed a systematic method:

(1) Calculate the onset times of all recorded taps using Onset Detection techniques from the `madmom` library [2].
(2) Match each onset to its nearest MIDI note and discard any onset that does not align with a MIDI note, assuming it results from a performance mismatch.
(3) Retain onsets that fall within a MIDI note's duration.
(4) Resolve cases where multiple onsets overlap a single MIDI note using a hierarchical priority: KD > SD > HH because it is the instrument class hierarchy present in the GMD dataset.

In the end, we ended up with a total of 9124 tap one-shots, of which 2273 were labeled as KD, 3883 as SD, and 2968 as HH.

4.3 Synthesis Process

4.3.1 Incorporating Velocity Information. Directly applying one-shot samples without incorporating velocity variation can result in monotonous and unrealistic drum sequences. However, one of the major advantages of the GMD dataset is the inclusion of MIDI velocity data (because the dataset was recorded directly by drummers on an electronic drum kit), which provides information about the intensity of each drum hit. To improve the realism of our synthetic examples, we mapped MIDI velocity values to a set of audio features extracted from our one-shot samples. Inspired by [27], we analyzed several features, including Peak Amplitude, Loudness, Sound Pressure Level (SPL), Spectral Centroid, Temporal Centroid and Spectral Flatness.

To determine which audio feature best mapped to velocity, we analyzed the distribution of MIDI velocities across all drum classes and computed each candidate feature for all one-shot samples, categorized by class. In order to compare velocity and feature distributions, we normalized the velocity values as well

as those of each feature to a range between [0, 1]. We then calculated the Wassertein-1 distance (or Earth Mover's Distance) between the MIDI velocity histograms and the corresponding feature distributions to assess their alignment. A total of 249737 MIDI notes were analyzed (KD: 158407, SD: 47194, HH: 44136) as well as all of the one-shots we recorded. The results are shown in Table 1.

As can be seen, we found that SPL offered the best mapping, preserving the dynamic range. SPL quantifies the perceived loudness of a sound, expressed in decibels (dB), by comparing the root mean square (RMS) pressure of the sound wave to a standard reference pressure. It can be mathematically defined as:

$$\mathrm{SPL} = 20 \log_{10} \left(\frac{\sqrt{\frac{1}{N} \sum_{i=1}^{N} x_i^2}}{p_{\mathrm{ref}}} \right)$$

where x_i denotes the audio samples, N is the total number of samples, and $p_{\mathrm{ref}} = 20 \times 10^{-6}$ Pa represents the reference sound pressure level in air (20 μPa), corresponding approximately to the threshold of human hearing. This ensures that higher-velocity MIDI notes correspond to louder, more forceful taps, preserving the intended dynamics of the performance.

We omit the other histograms for the sake of brevity, but in Figure 3 we can see the overlapping histograms for the normalized velocity and SPL values for the three classes.

4.3.2 Final Mapping Strategy. To work towards a natural and realistic tap sequence, we employed a systematic mapping strategy. First, both MIDI velocity values and SPL values were normalized to a range of [0,1], ensuring comparability. Then, for each MIDI event, we selected the one-shot sample whose SPL value was closest to the corresponding MIDI velocity, effectively performing a linear mapping between the two. To prevent unnatural silences that could manifest as gaps in the spectrogram, we ensured that the selected one-shot sample had a duration at least as long as the MIDI note it represented.

5 Evaluation

This section evaluates our proposed approach, focusing on model training and a comparison between different trained models. We assess how well our prototypical ADTS learns from the synthesized dataset and analyze its performance against an ADT model we trained on the GMD dataset, and another model trained on the small set of recorded tapped performances. Since there is no other ADTS system against which we can evaluate our own, conducting a standard evaluation process is challenging.

5.1 Model Training

5.1.1 Dataset Preparation. The resulting dataset consists of 315 examples, totaling 11.2 GB of data and approximately 8 hours and 47 minutes of audio. All audio files are uncompressed WAV format, 44.1 kHz sample rate, 64-bit resolution, mono. The dataset was split into training (80%), validation (10%), and test (10%) sets. For the Synthesized-GMD (S-GMD) dataset, we excluded all '*fill*' examples as they were too short and did not align with our hypothesis that the structure of drum patterns plays a crucial role.

5.1.2 Training Procedure. We chose a TCN [6] architecture due to its efficiency, parallelizability, and ability to account for temporal context. We used [15]'s `pytorch` implementation.

The input features were extracted using the `madmom`'s `LogarithmicFilteredSpectrogram` with the following parameters:

- Frame rate: 100 fps
- Minimum frequency: 30 Hz
- Maximum frequency: 15000 Hz
- Frame size: 2048
- Sample rate: 44100 Hz
- Bands per octave: 12
- Channels: mono
- Normalized filters: True

The model architecture consists of a TCN. The specifications were manually fine-tuned when training the model for ADT:

- Input channels: 79
- Hidden layers: [69, 59, 49, 39, 29, 19, 9, 3]
- Kernel size: 5
- Custom dilations: 1, 3, 9, 27, 81, 243, 729, 2187
- Dropout rate: 0.15
- Activation function: ELU
- Output projection: 3 (one per drum class)
- Output activation: Sigmoid
- Causal: False
- Input shape: NLC

Finally, for peak-picking, we used `madmom`'s implementation with the following parameters:

- Threshold: 0.05
- Smoothing: 0.0
- Pre-average window: 0.01s
- Post-average window: 0.01s
- Pre-max window: 0.02s
- Post-max window: 0.02s
- Combine window: 0.02s
- Frame rate: 100 fps

The dataset split ensured that the test set remained held out for final evaluation. We also applied early-stopping, with a patience of 10 epochs. The model training was conducted on an NVIDIA GA104M [GeForce RTX 3080 Mobile / Max-Q] (8GB/16GB VRAM), with an average training time of approximately one hour.

5.2 Results

In total, we trained the same TCN model (with the specifications above) on three different datasets: the original drums GMD dataset; our small collection of paired tapped recordings which we call our Quasi-Aligned (Q-A) dataset because of timing mismatches between the playing and the original drum patterns, and objective limitations of using only two hands while emulating a drumkit; and our S-GMD dataset, obtained by employing our strategy of section 4. In table 2, we can see how each of the trained models fares across datasets.

As expected, the models trained on tapped datasets fare poorly when trying to perform ADT and the model trained on GMD fares pretty well (F-Measure = 0.824). However, our results show that a model trained on a standard drums dataset cannot solve ADTS when applied directly to a tapped dataset, neither the live recordings nor the synthesized dataset. We can also see that the model trained on the S-GMD achieved good results on synthesized tapped sequences, achieving an F-Measure of 0.711. However, when considering real-world tap recordings, scores drop significantly. This attests to the difficulty of ADTS. The model trained on the Q-A performs poorly across the board, demonstrating that

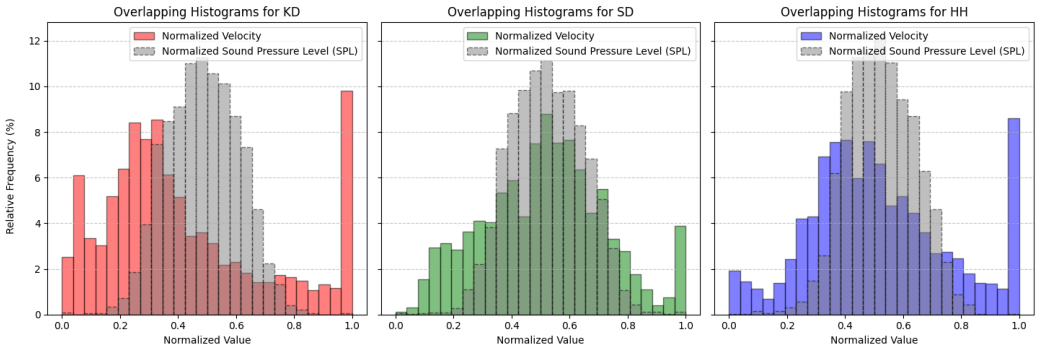

Figure 3: Overlapping histograms of the normalized velocity and SPL values for the three classes. We can see how SPL can be suitable for mapping velocity information.

Test dataset	GMD			Q-A			S-GMD		
Training dataset	GMD	Q-A	S-GMD	GMD	Q-A	S-GMD	GMD	Q-A	S-GMD
Precision	**0.862**	0.352	0.119	0.324	**0.390**	0.192	0.403	0.270	**0.657**
Recall	**0.868**	0.283	0.401	0.142	0.238	**0.439**	0.291	0.163	**0.908**
F-Measure	**0.824**	0.108	0.125	0.093	0.198	**0.206**	0.245	0.056	**0.711**

Table 2: Evaluation results for the TCN model using different dataset combinations. Q-A stands for "quasi-aligned dataset".

access to a large dataset is needed. This is evidenced when we directly compare the S-GMD model to the Q-A model: the Q-A model performs very poorly on the S-GMD dataset, whereas the S-GMD model performs adequately in this scenario. Conversely, even though the Q-A dataset improves performance on its own dataset, our model trained on only synthetic examples is on par and even surpasses the model that was trained specifically for that dataset, achieving the best F-Measure = 0.206. Considering we did not use any data augmentation techniques, which can help with the overall generalization of our model, these results show promise and warrant further research in our opinion.

6 Discussion and Conclusions

In this paper, we have introduced a new music analysis task: Automatic Drum TapScription (ADTS). It consists in obtaining a drum symbolic representation from arbitrarily-timbred percussive sounds. It is a novel and yet to be solved task, mainly because relative timbre differences are less noticeable than in traditional ADT and because of the shortage of data. To address this, we developed a strategy to synthesize a paired dataset between tapped versions of drum patterns and their symbolic counterparts from a collection of labeled one-shot taps, obtained by processing a small dataset of paired tapped recordings. The drums dataset used for the recordings and the synthesis was the GMD. Finally, we trained three TCN models, all with the same specifications, on three different datasets: the original GMD dataset, the small quasi-aligned tap recordings, and our synthetic tapped sequences (which we called S-GMD).

From the results, we can conclude that: 1) The model trained on the GMD (i.e. trained to solve the task of ADT) is ineffective for ADTS; 2) the model trained on our synthetic dataset has a much better performance on synthetic tapped pattern examples; 3) All models have poor performance on real-world tapped recordings, however, the model trained on the synthetic dataset exhibits the best performance. Essentially, each model overfits to the type of data it was trained on, reinforcing the challenge of generalizing to out-of-dataset tap timbres, as discussed in subsection 3.2. As for the reasons why this is so, we suspect that, since the one-shots used were directly processed from the same recordings, the synthesis process might still be producing hardly perceptible artifacts which might affect out-of-distribution inference.

In that sense, we believe that a great deal of improvements can be made, both to the dataset synthesis and the model definition and training process itself. First and foremost, data augmentation techniques such as reverb augmentation, consisting of simulating different acoustic environments; noise injection consisting of adding background noise to enhance generalization; pitch shifting consisting of introducing slight variations in pitch to mimic natural drum resonance; and temporal jittering consisting of slightly varying onset times to emulate human performance nuances may improve performance on in-the-wild examples. A percentile-based mapping between velocity and SPL can also improve dataset synthesis. Furthermore, performing hyperparameter fine-tuning should also help with generalizability. In addition to that, tweaking peak-picking thresholds and parameters, a crucial part of activation-based ADT systems, should enable more balanced scores between precision and recall. Finally, studying and applying data representations other than frames, such as beats, might help reduce the sparsity of the data and in turn improve the balance between recall and precision.

Beyond these improvements, we believe that further work is warranted, namely testing other DL architectures such as CNNs and comparing them with TCN models. Doing so would be able to test the hypothesis of the need for a larger temporal context for ADTS *vs* ADT. It could also be interesting to test how fast the inference can be and see if an ADTS model can be applied to real-time applications. Finally, having a reliable ADTS system can allow more precise and controllable percussive timbre transfer downstream.

7 Ethical Standards

This research did not involve human or animal experimentation. The tapped drum pattern recordings used in this study were performed voluntarily by the first author, without the participation

of external participants. No ethical concerns related to human or animal welfare apply. There are no conflicts of interest to declare.

Acknowledgments

This work is funded by the FCT - Fundation for Science and Technology (under grants SFRH / BD / 139775/2018 and SFRH / BD / 133415/2017), IPP / MCTES through national funds (PID-DAC), within the scope of CISUC R&D Unit—UIDB/00326/2020 or project code UIDP/00326/2020.

The first author is also funded by the FCT - Foundation for Science and Technology, under the grant 2021.05653.BD.

Finally, the collaboration between the first author and Roger B. Dannenberg from CMU was possible thanks to a Fulbright Scholarship.

References

[1] Juan P. Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. 2005. A Tutorial On Onset Detection In Music Signals. *IEEE Transactions on Speech and Audio Processing* 13, 5 (2005), 1035–1047. https://doi.org/10.1109/TSA.2005.851998

[2] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. 2016. Madmom: A New Python Audio And Music Signal Processing Library. In *Proceedings of the 24th ACM International Conference on Multimedia*. Amsterdam, 1174–1178. https://doi.org/10.1145/2964284.2973795

[3] Antoine Caillon and Philippe Esling. 2021. Rave: A Variational Autoencoder For Fast And High-quality Neural Audio Synthesis. https://doi.org/10.48550/arXiv.2111.05011

[4] Lee Callender, Curtis Hawthorne, and Jesse Engel. 2020. Improving Perceptual Quality of Drum Transcription with the Expanded Groove MIDI Dataset. https://doi.org/10.48550/arXiv.2004.00188

[5] Keunwoo Choi and Kyunghyun Cho. 2019. Deep Unsupervised Drum Transcription. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*. ISMIR, Delft, 183–191.

[6] Matthew E. P. Davies and Sebastian Böck. 2019. Temporal Convolutional Networks For Musical Audio Beat Tracking. In *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, Coruña, 1–5. https://doi.org/10.23919/EUSIPCO.2019.8902578

[7] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. 2019. Demucs: Deep Extractor For Music Sources With Extra Unlabeled Data Remixed. https://doi.org/10.48550/arXiv.1909.01174

[8] Jon Gillick, Adam Roberts, Jesse Engel, Douglas Eck, and David Bamman. 2019. Learning to Groove with Inverse Sequence Transformations. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 2269–2279. https://proceedings.mlr.press/v97/gillick19a.html

[9] François Haas, Suzan Distenfeld, and Kenneth Axen. 1986. Effects Of Perceived Musical Rhythm On Respiratory Pattern. *Journal of applied physiology* 61, 3 (1986), 1185–1191.

[10] Behzad Haki, Teresa Pelinski, Marina Nieto, and Sergi Jorda. 2023. Completing Audio Drum Loops with Symbolic Drum Suggestions. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. NIME, Mexico City, 236–243.

[11] Thomas Holz. 2022. *Automatic Drum Transcription with Deep Neural Networks*. Master's thesis. Technische Universität Berlin - TU Berlin.

[12] Ryoto Ishizuka, Ryo Nishikimi, and Kazuyoshi Yoshii. 2021. Global Structure-Aware Drum Transcription Based on Self-Attention Mechanisms. *Signals* 2, 3 (2021), 508–526. https://doi.org/10.3390/signals2030031

[13] John Iversen. 2016. *In the Beginning Was the Beat: Evolutionary Origins of Musical Rhythm in Humans*. https://doi.org/10.1017/CBO9781316145074.022

[14] Céline Jacques and Axel Roebel. 2018. Automatic Drum Transcription With Convolutional Neural Networks. In *Proceedings of the 21th International Conference on Digital Audio Effects*. DAFx, Aveiro, 80–86.

[15] Paul Krug. 2023. PyTorch Temporal Convolutional Networks. https://github.com/paul-krug/pytorch-tcn Accessed: 2025-02-04.

[16] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. 2023. High-Fidelity Audio Compression with Improved RVQGAN. In *Advances in Neural Information Processing Systems*, Vol. 36. Curran Associates, Inc., 27980–27993. https://proceedings.neurips.cc/paper_files/paper/2023/file/58d0e78cf042af5876e12661087bea12-Paper-Conference.pdf

[17] Matz Larsson. 2014. Self-generated Sounds Of Locomotion And Ventilation And The Evolution Of Human Rhythmic Abilities. *Animal cognition* 17, 1 (2014), 1–14.

[18] Matz Larsson, Joachim Richter, and Andrea Ravignani. 2019. Bipedal Steps in the Development of Rhythmic Behavior in Humans. *Music & Science* 2 (2019). https://doi.org/10.1177/2059204319892617

[19] Daniel J. Levitin, Jessica A. Grahn, and Justin London. 2018. The Psychology Of Music: Rhythm And Movement. *Annual review of psychology* 69, 1 (2018), 51–75.

[20] Rémi Mignot and Geoffroy Peeters. 2019. An Analysis of the Effect of Data Augmentation Methods: Experiments for a Musical Genre Classification Task. *Transactions of the International Society for Music Information Retrieval* 2 (2019), 97–110. https://doi.org/10.5334/tismir.26

[21] Thomas Nuttall, Behzad Haki, and Sergi Jorda. 2021. Transformer Neural Networks for Automated Rhythm Generation. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. NIME, Shanghai. https://doi.org/10.21428/92fbeb44.fe9a0d82

[22] Patrick O'Reilly, Hugo Flores Garcia, Prem Seetharaman, and Bryan Pardo. 2024. Masked Token Modeling for Zero-Shot Anything-to-Drums Conversion. In *Extended Abstracts for the Late-Breaking Demo Session of the 25th International Society for Music Information Retrieval Conference*. ISMIR, San Francisco.

[23] Chih-Wei Wu Richard Vogl, Carl Southall. 2017. Drum Transcription - MIREX Wiki. https://www.music-ir.org/mirex/wiki/2017:Drum_Transcription Acessed: 2025-04-15.

[24] Rohit M A, Amitrajit Bhattacharjee, and Preeti Rao. 2021. Four-way Classification of Tabla Strokes with Models Adapted from Automatic Drum Transcription. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, 19–26. https://doi.org/10.5281/zenodo.5624489

[25] André C. Santos and F. Amilcar Cardoso. 2023. From Taps to Drums: Audio-to-audio Percussion Style Transfer. In *Extended Abstracts for the Late-Breaking Demo Session of the 24th International Society for Music Information Retrieval Conference*. ISMIR, Milan.

[26] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction To Information Retrieval*. Vol. 39. Cambridge University Press Cambridge.

[27] Jordie Shier, Charalampos Saitis, Andrew Robertson, and Andrew McPherson. 2024. Real-time Timbre Remapping With Differentiable Dsp. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. NIME, Utrecht.

[28] Carl Southall, Ryan Stables, and Jason Hockman. 2016. Automatic Drum Transcription Using Bi-Directional Recurrent Neural Networks. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*. ISMIR, New York City, 591–597.

[29] Carl Southall, Ryan Stables, and Jason Hockman. 2017. Automatic Drum Transcription For Polyphonic Recordings Using Soft Attention Mechanisms And Convolutional Neural Networks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*. ISMIR, Suzhou.

[30] Koen Tanghe. 2005. Audio Drum Detection - MIREX Wiki. https://www.music-ir.org/mirex/wiki/2005:Audio_Drum_Det Acessed: 2025-04-15.

[31] Richard Vogl, Matthias Dorfer, and Peter Knees. 2017. Drum Transcription From Polyphonic Music With Recurrent Neural Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, ICASSP, New Orleans, 201–205.

[32] Richard Vogl, Matthias Dorfer, Gerhard Widmer, and Peter Knees. 2017. Drum Transcription via Joint Beat and Drum Modeling Using Convolutional Recurrent Neural Networks.. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*. ISMIR, Sezhou, 150–157.

[33] Richard Vogl and Chih-Wei Wu. 2019. Drum Transcription - MIREX Wiki. https://www.music-ir.org/mirex/wiki/2019:Drum_Transcription Acessed: 2025-02-04.

[34] Yu Wang, Justin Salamon, Mark Cartwright, Nicholas J Bryan, and Juan Pablo Bello. 2020. Few-shot Drum Transcription In Polyphonic Music. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*. ISMIR, 117–124.

[35] 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