# The Rhythm In Anything

Patrick O'Reilly, Julia Barnett, Hugo Flores Garcia, Annie Chu,
Nathan Pruyne, Prem Seetharaman, Bryan Pardo

**Index Terms**—Music Information Retrieval, Audio Generation, Drum Synthesis, Artificial Intelligence

## 1 Introduction

The Rhythm In Anything, or TRIA, is an artificial intelligence model trained to generate drums with audio-prompted rhythm and timbre. The work from Patrick O'Reilly et al. fall within Music Information Retrieval discipline, which is the science of retrieving information from music. The novelty here is the high fidelity drum recordings, generated from a rhythm and a timbre audio. The model takes simple rhythm patterns, an audio of some beatboxing or tapping sounds; and as drumkit timbres, it only takes an example recording. The strength of this model is not only the high fidelity of the audio generated, but also that it works even with timbres it was not trained on. The article focuses on how the dualization of rhythm and timbre allows the model to give better synthetizations. The code is fully available on their github and webpage.

Here, the authors are presenting a solution for high-quality audio generation model able to capture rhythmic information contained in audio gestures, which will be used to generate drum audio with controllable timbre.

## 2 State of the Art - Before Publication

The progress of Artificial Intelligence lead to rapid innovation in all branches to which artificial intelligence can be applied to. Including AI-driven music generation. New models emerge, targetting different aspects of musical structure, with for instance lyrics transcription, instrument separation, and audio generation. The Rhythm In Anything [1] can be contextualized in music generation, more specifically, audio-prompted drum generation.

At the time of the publishing of this article, MelodyFlow by Lan et al. [2] is the most recent model with a lot of influence in music generation models. It is a model released in 2024, which was made for music edition and generation. Based on text descriptions and an audio input, it generates an audio which is based on the audio input, with the modifications specified in the text prompt. This process of keeping close to the original audio is called flow matching. The music samples work in stereo, and the audio generated is returned directly in waveform latent space. The high quality of the generations as well as its controllability make MelodyFlow a very good model to which the authors can compare their own model. This model will be used with rhythm audio prompts, and the timbre specification will be text prompted.

The authors are also referring to GrooVAE models [4]. GrooVAE is a class of models which are designed to generate full drummkit expressive performances. The input here is a symbolic rhythm representation, more specifically a single voice MIDI file of drum patterns. GrooVAE models are able to reconstruct or generate an audio with micro-timings and velocity variations. The novelty brought by GrooVAE in 2019 is the mapping of symbolic rhythm to a full performance. This had a big influence in Music Information Retreival and music generation because it shows the importance of timings and dynamics.

RAVE [5] is an other model the authors took their inspiration from. RAVE stands for Real-time Audio Variational autoEncoder. It is a neural audio generative model from 2023, based on variational autoencoders which are used in AI models for generation. Variational autoencoders are pre-trained models which learn a probability distribution in the latent space, and are used to train other generative models. In RAVE, variational autoencoders allow the model to learn from the waveform in real-time. This makes RAVE a powerful model with fast generation while allowing for explicit latent control over timbre and quality. In this context of rhythm pattern generation, RAVE models have been used on tap-to-drum translation into audio.

Beyond these core models, which served as main inspiration source for the authors, other audio generation models were proposed, which take rhythmic structure into account. However, the same constraints and problems arise in all those models. The previous propositions and grooVAE among them are not taking audio specification of timbre nor of rhythm. Most of them, including RAVE, need re-training for a different timbre or specifications not learnt in the training phase. The user needs to calibrate specifically for better transcriptions. Often, limitations are seen in the sound-gestures types and the litteral mapping of timbres. And finally, we only see dualization in the symbolic domain in models for music generation.

## 3 Contribution

The authors partition their contribution into three separate parts. First, the model, then the dualization task, and finally how the evaluations carried out show the performances of the model.

We will first discuss here the creation of the dataset used, and the specifications for the experiments. Then we will talk about the core of the article: what TRIA is able to do. Finally we will detail the proposed methodology, and present the experiments and results.

### 3.1 Datasets

TRIA was trained and evaluated on audio rhythm prompts taken from two datasets: Amateur Vocal Percussion - AVP [7], and TapTamDrum [6], as well as MoisesDB dataset for timbre prompts.

AVP is a dataset compiling beatboxing improvization recordings from twenty-eight amateurs. The sound gestures expressed in beatboxing and gathered in AVP are varied and extensive. AVP's samples are also fully annotated.

TapTamDrum is a dataset composed of rhythms imitations, reproduced by tapping on surfaces. This gives a temporal structure stronger than the timbral aspect.

Fifty-six beatbox samples were taken from AVP dataset, as well as one thousand one hundread and sixteen tapping samples from TapTamDrum dataset, in order to form the rhythm prompt dataset for TRIA's training and evaluation.

MoisesDB is a public dataset of professional quality music stems extracted from real, multitrack, recordings. TRIA's timbre prompts are small snippets taken from MoisesDB's drum tracks. The high quality of the MoisesDB samples gives TRIA a strong base for its generations.

For the experimentation part. Since MelodyFlow is the method used for comparison, the authors had to adapt and translate the audio timbre prompt into a suitable text prompt. To do so, the autors used Chat-GPT 4.5 to generate fifty descriptions of drumkits. The text prompts were all inspected manually, and the authors also consulted with MelodyFlow's authors to ensure the reliability of the generated prompts. Excerpts form MoisesDB were also used in the experimentation phase, to compare the audio quality of TRIA's generations to real drum recordings.

### 3.2 Tasks

TRIA is designed to perform audio generation of drums. Given two audio prompts - one of the desired rhythmic pattern, the other a sample describing the desired drum timbre - TRIA generates a full drum recording that plays the requested rhythm in the given timbre. The model is trained as a masked transformer that predicts missing audio tokens conditioned on both rhythm and timbre information, enabling it to translate arbitrary rhythmic sound gestures into high-fidelity drum audio in a zero-shot manner.

### 3.3 Methodology

TRIA is a transformer-based masked language model. A transformer-based model is a neural network which relies on self-attention mechanisms that allow the network to model long-range dependencies within a sequence by relating each element to all others in parallel. TRIA is also a masked model, which trains the model to predict missing tokens. Tokenization is a process of breaking down input data into a sequence of tokens that the model can process. Here the input is an audio file, thus the tokens are short sequences of audio. At training time, some tokens are randomly hidden, and the model has to correctly retreive the audio token using context from unmasked tokens. Before tokenization, audio is buffered into segments of fixed length. Audio is then encoded using a Differentiable Audio Codec (DAC). This buffering ensures temporal continuity and allows the model to capture rhythmic patterns across longer time spans.

The second main part of this work is the dualization, or more globally, the splitting of the rhythm prompt's spectrogram into several parts. The splitting works as follows: First the authors acquire the spectrogram from the audio rhythm prompt. Then, the authors look at the amplitude in frequencies, and determine where the splitting must happen. The energy needs to be balanced across the bands to be able to distinguish between low and high frequencies and better capture rhythmic information. Finally, for each band created, the energy is summed to extract rhythmic activity.

The authors tried different versions, with a different number of bands to try and capture rhythm more or less precisely, and see a difference in the model's performances. The aim is to encode well the rhythmic structure given in input and avoid leaking timbre information. This results in the creation of different versions of TRIA: TRIA 1-band, where no separation is performed. TRIA 2-band, which will be the reference version, is adaptative and splits the spectrum in two bands. TRIA 2-band non-adaptative splits the spectrum in the middle, there is no adaptative alignment in this version so it is less effective in most cases. TRIA 3-band and 4-band are splitting the spectrogram into respectively three and four.

The state of the art model used is MelodyFlow, which takes in input an audio prompt of the desired rhythm - sond gestures just like for TRIA, and a text prompt of the desired timbre. MelodyFlow is a one billion parameter transformer model. It was trained both on public and private data, for twenty thousand hours of music in total. There is a parameter that controlls how much the generated audio keeps the rhythmic structure of the given rhythm prompt. This target flow step is a parameter that ranges from 0.0 to 1.0, corresponding respectively to full noising and no noising. In other words, values closer to 0.0 will lower the influence of the rhythm prompt in the output audio The authors chose to use values 0.0, 0.1, and 0.2 for this target flow step parameter, they found that greater values resulted in synthetizations that did not take the timbre prompt into account enough. This results in three versions, used for comparison in the experimentation, respectively Melodyflow_0.0, Melodyflow_0.1, and Melodyflow_0.2.

### 3.4 Experiments and Results

The authors conducted subjective and objective evaluations to assess the performances of TRIA. They wanted to measure the quality of the synthetizations, as well as how close the audio generated was to its audio-prompted timbre and rhythm.

#### 3.4.1 Subjective Evaluation

The subjective evaluation was carried out in order to verify how musically pleasing the generated audios were, and also how the synthetizations from TRIA compared to MelodyFlow's generations and the random excerpts. For comparison purposes, in these evaluations the authors compare audio generated from the model TRIA 2-band with random audios extracted from the dataset MoisesDB, and also with audio generated from the model MelodyFlow 0.2. Evaluators here are humans, recruited through the platfomr Prolific. The evaluations were done through ReSEval, which stands for Reproducible Subjective Evaluation. Which is a framework used for building subjective evaluations. In order to have an homogeneous testers base, the people recruited had to pass a listening test. Out of the a hundread and twenty persons originally recruited, a hundread and sixteen passed the listening test and went on with the evaluation of the audios. For the subjective evaluation, listeners rated audio from eighty sets. A set is composed of an audio rhythm prompt, the associated generation from TRIA 2-band, and from MelodyFlow 0.2, and a drum extract randomly taken from MoisesDB. The eighty generations were made with ten rhythm prompts: five with tapping sounds, and five with beatboxing audio, on eight different audio timbre prompts. Each audio clip generated lasts for three to four seconds, which is the duration of the given rhythm prompt. Three pairwise comparisons were evaluated for each set by five persons, comparing TRIA to MelodyFlow, Tria to the random excerpt, and MelodyFlow to the random excerpt. Each listener was given ten pairwise comparisons to evaluate.

The subjective evaluation shows there are no clear preference when comparing audio synthetized from TRIA 2-band or Melodyflow 0.2. However, when comparing audio generated from either of these models to the random extract from the Moises database, the favourite one is most often the generated audio.

This is very promising. First, because it means that generations from TRIA are on par with generations from MelodyFlow, which indicates that the authors were able to build a model that performs at least as well as the primary existing model. Moreover, TRIA is a very small model compared to MelodyFlow. MelodyFlow is a model twenty-five times the size of TRIA, and was trained on two thousand times more data. Finally, being able to have generations well liked by human listeners but, most of all, with audio input is an even greater achievement. Allowing for audio prompting makes the creative process a lot easier than the complex task of translating the audio prompts into a text that will be correctly interpreted by the model.

### *3.4.2 Objective Evaluations*

The objective evaluations are conducted according to three criteria: adherence to the rhythm prompt, adherence to the timbre prompt, and audio quality of the synthesized outputs.

The evaluation of the adherence to the rhythm prompt aims to measure how well the rhytmic structure of the rhythm prompt was preserved in the associated generation. Evaluations were carried out on all variants of TRIA and MelodyFlow, and on the random anchor. The metric used here is the F1 score, to measure the correspondance between transcription - the audio generated by the model, and ground truth - the prompt given to the model in input, supplemented with human annotations about the kick, snare, and hi-hat in the audio. To compute the F1 score, both the generated audio and the rhythm prompt are transcribed into discrete drum event sequences. For each of the sequences, the precision measures the proportion of correctly generated events among all predicted events, while recall measures the proportion of ground-truth events that are successfully repreduced in the generated output. The F1 score is the harmonic mean of precision and recall. A higher F1 score indicates a tighter correspondence between the rhythmic structure of the prompt and that of the generated audio.

Adherence to the timbre prompt evaluates how well the instrumental characteristics from the timbre audio prompt are preserved in the synthesized output. In TRIA, timbre prompts consist of short drum audio excerpts drawn from MoisesDB, and the evaluation assesses whether the generated audio reflects the spectral and perceptual qualities of the provided timbre example. Unlike rhythm adherence, timbre similarity is evaluated using audio embedding distances. The generated audio is compared to the timbre prompt in a learned feature space designed to capture perceptual similarity.

Finally, the realism and perceptual quality of the synthetized audio are evaluated by comparing TRIA's outputs to random drum excerpts from MoisesDB, which serve as a reference distribution of real drum sounds. Here the authors decided to avoid comparing TRIA's generations against MelodyFlows since MelodyFlows uses text prompt describing the timbre prompt. The metric used here is Kernel Audio Distance, or KAD. KAD measures the distance between two distributions of audio samples by computing a Maximum Mean Discrepancy (MMD) between their embeddings in a learned feature space. Lower KAD values indicate that the generated audio is closer to the reference distribution and therefore more realistic. The authors used two types of KAD for the evalutation: KAD-PANN, relies on embessings extracted from a Pretrained Audio Neural Network model, emphasizing acoustic and timbral characteristics. And also KAD-ClapLaionMusic, which uses embeddings from a CLAP model trained on the LAION-Music dataset. It captures higher-level perceptual and musical attributes, from a human-like point of view. The authors used both methods to assess audio quality from complementary perceptual perspectives.

The results of the rhythmic evaluation show TRIA 2-band is performing significantly better than MelodyFlow for beatboxing. The kick and snare placement is better matching the audio prompt. That is mostly due to the dualization of the audio prompt. TRIA 1-band and 2-band non adaptive are overtaken by TRIA 2-band. Whereas 3-band and 4-band versions of TRIA only show slightly better results in the kick placement, and not so much improvement for the snare placement. Here, the authors conclude that in order to capture the rhytmic structure of an audio, splitting the spectrogram in two may be efficient enough, to get a synthetization really close to the rhythm audio prompted.

Then the results of the timbre evaluation are demonstrating that TRIA's generations give a lower spectral correlation with the timbre prompt, and a higher correlation with the timbre prompt. whereas MelodyFlow's generations show a higher spectral correlation with the rhythm prompt, which attests for timbre leakage. The timbre evaluation highlights TRIA's ability to recombine rhythm and timbre in a zero-shot manner. This is not available in text-guided models such as MelodyFlow.

The authors finish by concluding that TRIA is outperforming MelodyFlow since TRIA's generations are more precise and close to the prompted rhythm and timbre, thanks to the inputs being given as audios, and the dualization of the spectrograms. These, combined with TRIA's zero-shot timbre specification, are TRIA's strongest advantages.

## 4 Discussion

## 5 State of the Art - After Publication

## References

[1] P. O'Reilly, J. Barnett, H. F. Garcia, A. Chu, N. Pruyne, P. Seetharaman, and B. Pardo, "The Rhythm In Anything: Audio-Prompted Drums Generation with Masked Language Modeling," in *Proc. of the 26th Int. Society for Music Information Retrieval Conf.*, Daejeon, South Korea, 2025.

[2] G. L. Lan, B. Shi, Z. Ni, S. Srinivasan, A. Kumar, B. Ellis, D. Kant, V. Nagaraja, E. Chang, W.-N. Hsu, Y. Shi, and V. Chandra, "High fidelity text-guided music editing via single-stage flow matching," *arXiv preprint arXiv:2407.03648*, 2024.

[3] I. Pereira, F. Araújo, F. Korzeniowski, and R. Vogl, "Moisesdb: A dataset for source separation beyond 4-stems," in *Conference of the International Society for Music Information Retrieval (ISMIR)*, 2023.

[4] J. Gillick, A. Roberts, J. Engel, D. Eck, and D. Bamman, "Learning to groove with inverse sequence transformations," in *ICML*, 2019.

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

[6] B. Haki, B. Kotowski, C. Lee, and S. Jorda, "Taptamdrum: A dataset for dualized drum patterns," in *Proceedings of the 24th International Society for Music Information Retrieval Conference*, ISMIR, 2023.

[7] A. Delgado, S. McDonald, N. Xu, and M. Sandler, "A new dataset for amateur vocal percussion analysis," in *Proceedings of the 14th International Audio Mostly Conference: A Journey in Sound*, ser. AM '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 17–23. [Online]. Available: https://doi.org/10.1145/3356590.3356844