BEAT-ALIGNED SPECTROGRAM-TO-SEQUENCE GENERATION OF RHYTHM-GAME CHARTS

Jayeon Yi
University of Michigan
jayeonyi@umich.edu

Sungho Lee
Seoul National University
sh-lee@snu.ac.kr

Kyogu Lee
Seoul National University
kglee@snu.ac.kr

ABSTRACT

In the heart of "rhythm games" - games where players must perform actions in sync with a piece of music - are "charts", the directives to be given to players. We newly formulate chart generation as a sequence generation task and train a Transformer using a large dataset. We also introduce tempo-informed preprocessing and training procedures, some of which are suggested to be integral for a successful training. Our model is found to outperform the baselines on a large dataset, and is also found to benefit from pretraining and finetuning.

1. INTRODUCTION

Rhythm games - games where players perform specific actions in sync with a piece of music - are enjoyed by millions of players worldwide [1]. A number of past researches have investigated the automated design of the sequence of actions, or *charts*, given the music [1–3]. As the ground truth depends on the desired "difficulty" of the chart as well as the music, the past approaches fed spectrograms and a encoded difficulty parameter to deep neural networks (DNNs), inferring if each "audio frame" contains a chart event. However, such approaches were observed to exhibit binary class imbalance due to the temporal sparsity of ground-truth chart events, especially for low-difficulty charts [1]. As a remedy Takada *et al.* [3] successfully proposed to incorporate tempo information, but the sparsity itself persists.

We newly formulate chart generation as a conditional sequence generation task, thus removing the binary class imbalance. We beat-align and normalize the length of our four-beat training samples, train a Transformer [4] on our large dataset, and fine-tune to the common benchmarks. We also examine how our approach scales with more data, and how our beat-aligned training procedures may be integral to a successful training. Our method outperforms previous approaches in terms of micro-F1 scores. We share demos, code, and pretrained models on our companion
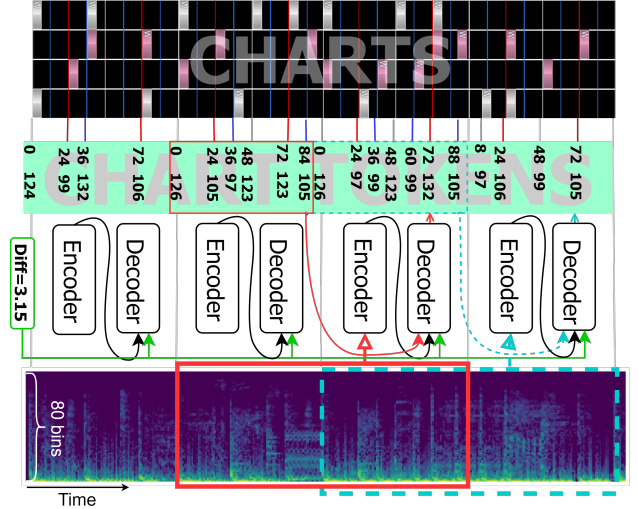
Figure 1. A schematic of our approach. Four beats of beat-aligned log-Mel spectrograms, a difficulty value, and *chart tokens* for the two former beats are used to infer chart tokens for the two latter beats.

website [1] .

2. TEMPO-INFORMED PREPROCESSING

2.1 Dataset

To facilitate training, we build a large dataset on the popular "four-keys"(4k) mode of "osu!mania", a game where players follow on-screen instructions to tap, hold, or stop holding each of the four "keys" in sync with a piece of accompanying music. We collected a total of 14648 charts aligned with a total of 3166 stereo-audio songs, the alignment of which were peer-reviewed by seasoned community creators. 6781 charts and 2004 songs remained after only leaving "4k" charts without off-beat tempo changes or extremely dense (>25 events/beat) parts. We split the dataset into three splits as represented in Table 1, and analyze the frequency of various beat groups in Table 3.

The Fraxtil and ITG datasets [1] were similarly preprocessed for benchmarking purposes. These datasets are based on a game with a syntactically same set of controls as "osu!mania 4k". However, while the former provides algorithmic chart difficulty assessments as a floating-point number, the latter provides human-labeled integers.

---

[1] https://stet-stet.github.io/goct

Split	Train	Valid	Test
# songs	1598(73 hrs)	199(9.3 hrs)	207(9.8 hrs)
# charts	5494(219 hrs)	634(26 hrs)	643(25 hrs)
# beats	2.24M	252k	255k

Table 1. A summary of our "osu!mania" 4-key dataset. "# beats" are equivalent to the number of samples in each set, i.e. 2.24M samples were used for training. This dataset is approximately 10x bigger than Fraxtil and ITG datasets.

Method	Fraxtil (%)	ITG (%)	osu! (%)
CNN [1]	75.0	71.9	83.0
C-LSTM [1]	75.6	72.1	83.9
Ours	70.5	68.7	**84.6**
Ours (pretrained)	**78.9**	**74.3**	-

Table 2. F_1 scores on Fraxtil, ITG, and osu!mania datasets. Pretraining was done on the "osu!mania" dataset.

2.2 Tempo-Informed Preprocessing

Our new preprocessing procedures involve the beat-aligning of training samples - a classic idea in MIR literature (e.g. [5, 6]). We processed all music into mono-channel log-Mel spectrograms with 80 Mel bins, FFT and window sizes of 512, and hop sizes corresponding to 1/48 of a beat at any given moment. Thus, a piece of music with a length of 400 beats, regardless of tempo, would be processed into a matrix $S \in \mathbb{R}^{19200 \times 80}$.

2.3 Tokenization

We use *chart tokens* to formulate chart generation as a sequence generation problem. An eight-beat example chart is translated into chart tokens on the top of Fig. 1. Each chart event is expressed in two tokens - a *time token* (0-95) denoting its temporal position, and an *action token* (97-176) identifying which of the 80 acceptable player actions are associated with the event. A "separator" token can be used to set a reference for time tokens. Finally, The end-of-sequence (EOS) token is used to pad the sequence when needed. Chart tokens resemble MIDI tokens in symbolic music literature [7, 8], but are much simpler due to the narrower variety of chart events compared to MIDI.

3. GENERATING CHARTS

3.1 Model Architecture and Pipeline

We use encoder-decoder Transformers, which have enjoyed considerable success in musical sequence generation tasks [7, 8]. Our pipeline is depicted in Fig. 1. Four beats of log-Mel spectrograms are fed to the encoder, while the last seven chart tokens from the former two beats are concatenated with 48-dim difficulty value embeddings and fed to the decoder. Chart tokens for the latter two beats are autoregressively generated, and then similarly truncated and fed into the model again to produce tokens for the two beats to follow after. This continues until the end of the

Beat Groups	8th	16th	12th	32nd	24th
% freq	31.3	20.7	3.1	1.8	1.3
F_1 unaligned (%)	70.0	47.6	9.2	3.27	3.94
F_1 aligned (%)	**87.9**	**64.0**	**59.2**	**21.5**	**26.3**

Table 3. Micro-F_1 scores of aligned and unaligned models on "osu!mania" dataset for each timing group. Frequency of occurrence in dataset is denoted on the second row.

music is reached. We use three layers and an input dimension of 256, since larger models did not yield any increase in performance.

3.2 Training

For the proposed model, all training and validation samples are aligned to beat boundaries. We trained on each dataset for 10 epochs with the adam [9] optimizer with lr=2e-4, and bs=32. We used the cross-entropy loss with a label-smoothing of 0.02. We then took this model, "pretrained" on the "osu!mania" dataset, and then finetuned on the Fraxtil and ITG datasets for 4 epochs, with lr=2e-5.

Models compared to the baselines were trained *without action tokens*; the baseline models only infer temporal positions of events. All others are trained with action tokens.

3.3 Evaluation, Metrics and Analysis

Micro-F_1 scores, calculated from temporal locations of chart events, are presented on Table 2. While our approach falls short of the baselines on small datasets such as Fraxtil and ITG, it outperforms the baselines when trained on a large dataset such as "osu!mania". Moreover, pretraining on a large dataset significantly improves the scores. The results are reminiscent of past works which showed that large-scale training with a Transformer can trump inductive bias [10]. We conjecture that utilizing more data would help to further improve our approach.

Our approach is not without limitations - even the finetuned model fails catastrophically on generating low-difficulty charts of Fraxtil and ITG datasets. This is conjectured to be due to how "osu!mania" dataset rarely includes *very* sparse charts like these. We conjecture that training on a more diverse set of data could be a solution.

3.4 Ablation: Unaligned Training

We train a model on the "osu!mania" dataset using non-beat-aligned four-beat-long training samples, keeping all other conditions the same. Table 3 shows micro-F_1 scores of both models for moderately-used time-tokens, grouped by their temporal offset relative to beat boundaries. Similar to [1], any time-token from the unaligned model was deemed correct if within 30ms of any ground-truth time-token. Table 3 shows that the aligned model exhibits stronger generalization. Moreover, It was qualitatively observed that the unaligned model would output temporally shifted outputs, with the shift changing over time. This result suggests that the beat alignment is vital to the successful training of our transformer-based models.

4. REFERENCES

[1] C. Donahue, Z. C. Lipton, and J. McAuley, "Dance dance convolution," in *Proc. of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017, pp. 1039–1048.

[2] E. Halina and M. Guzdial, "Taikonation: Patterning-focused chart generation for rhythm action games," in *Proc. of the 16th International Conference on the Foundations of Digital Games*, 2021, pp. 1–10.

[3] A. Takada, D. Yamazaki, Y. Yoshida, N. Ganbat, T. Shimotomai, N. Hamada, L. Liu, T. Yamamoto, and D. Sakurai, "Genélive! generating rhythm actions in love live!" in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 5266–5275.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] D. P. Ellis and B.-M. Thierry, "Large-scale cover song recognition using the 2d fourier transform magnitude," 2012.

[6] S. Dieleman, P. Brakel, and B. Schrauwen, "Audio-based music classification with a pretrained convolutional network," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR-2011)*. University of Miami, 2011, pp. 669–674.

[7] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, "Mt3: Multi-task multitrack music transcription," in *Proc. of the 10th International Conference on Learning Representations*, 2022.

[8] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.