The Interface for Symbolic Music Loop Generation Conditioned on Musical Metadata

Sangjun Han
LG AI Research
sj.han@lgresearch.ai

Hyeongrae Ihm
LG AI Research
hrim@lgresearch.ai

Woohyung Lim
LG AI Research
w.lim@lgresearch.ai

Abstract

We develop a web interface that generates multi-track music loop sequences. Our system takes musical metadata from users as input conditions and generates MIDI token events that can be played seamlessly. The core component, the loop generation model, is trained with loop sets that have been extracted by observing the repetitive structure of music. Also, the metadata tokens are randomly dropped to ensure flexible controllability during training. Our interface is available at `https://github.com/sjhan91/loop-demo`.

1 Introduction

To realize human creativity in artworks, the interaction between humans and AI plays a crucial role in the context of generative models. Through an interactive system, humans can inspire AI to produce novel creations, and vice versa, empowering synergy for each other. Thus, it is requested to establish a straightforward interface that enables us to communicate with AI.

Music composition is a notably creative process in that it turns invisible concepts into the tangible form of sound. As did in recent works for other fields, music can be also generated using deep generative models, conditioned on mediums conveying ideas [1–5]. For symbolic music generation, however, there are two challenges to be addressed; 1) long token lengths in multi-track compositions, 2) the absence of well-established datasets containing conditions and their corresponding outputs. These challenges pose limitations on the training of conditional large-scale models.

Here, we leverage the repetitive nature of music by extracting and training loops, which serve as basic building blocks. Also, as did in [6, 7], we employ musical metadata as input conditions, which are extracted readily from the original MIDI. In other words, our generation model receives users' inputs (instruments, mean pitch, mean tempo, mean velocity, mean duration, and chords) and generates multi-track loop sequences that can be played seamlessly. Furthermore, we develop a web interface that users can easily manipulate. We hope that this serves as a meaningful attempt in bridging the gap for music between humans and AI.

2 Method

We introduce the process of preparing loop datasets for training autoregressive models. Then, we provide detailed explanations of the model description and training details in turn. Finally, we present our web interface along with its available features.

Data Preparation To accommodate various genres and tracks, we choose Lakh MIDI Dataset (LMD) [8] and MetaMIDI Dataset (MMD) [9] for both our training and evaluation sets. They are converted to a sequence of token events represented by REMI+ [6] to handle multiple tracks. We utilize similar configurations for REMI+ as explained in [6], only focusing on music with a 4/4 time signature.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

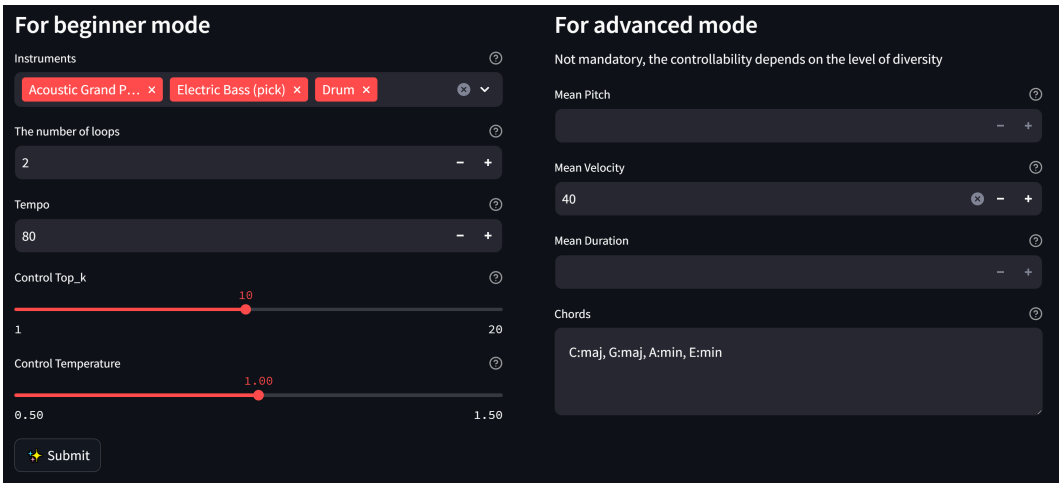

Figure 1: Loop Generation Web Interface

Loop Extraction The whole process of the loop extraction can be outlined as follows; extracting bar-level embeddings and discovering the repetitive structures in music. First, the bar-level embeddings for REMI+ events are extracted from customized BERT which is trained to predict masked tokens and to maximize the distance among bars within the same music. This implies preserving musical information in the embeddings and extracting distinctive features for each bar. Second, we construct a self-similarity matrix that captures bar relationships using the extracted embeddings. By transforming the self-similarity matrix to a time-lag matrix, we can identify recurring segments indicated by vertical lines at specific lag points. A comprehensive explanation of the procedure is provided in [10]. We extract those loop segments whose length is a multiple of 4 and less than 16. As a result, we acquire a total of 940M loop tokens, which have been compressed through music byte-pair encoding [11] to reduce overall token lengths ($\approx 40\%$ reduction).

Loop Generation Our generation model is based on a decoder-only autoregressive Transformer architecture, following **GPT-3 Medium** scale (313M) [12]. At the training phase, all metadata tokens (instruments, mean pitch, mean tempo...) and loop tokens are concatenated and engaged in the next token prediction. During training, we introduce random drops for each musical component of the metadata tokens to enhance flexibility in control. In other words, users are not required to complete all musical conditions to generate consistent loop sequences. This approach moderates the relationship among input conditions from "and" to "or".

Web Interface Our web interface built on Streamlit is user-friendly and straightforward to follow (Figure 1). For beginner mode, users have the options to select instrument sets, specify the number of loop repetitions, set the tempo, and adjust sampling diversity (top-k sampling and temperature). For advanced mode, users can specify the value of mean pitch, mean velocity, mean duration, and the sequence of chords. These advanced options are not mandatory and their impacts as conditional inputs diminish as the degree of sampling diversity increases.

3 Evaluation

We evaluate our generative model in terms of its model capacity, sample fidelity and diversity, and controllability. The details of our evaluation procedure and the result table are described in Appendix A. When the model is trained with the incorporation of random drops, we have verified that the performance remains consistent even when the subsets of conditions are injected. Without this, the performance across all metrics significantly degrades when partial conditions are given. It implies that non-experts do not need to struggle to complete all musical conditions.

4 Ethical Implications

Music is expected to demand more fidelity and creativity than writing. Originally, the two properties are in a trade-off relationship. As we place a greater emphasis on fidelity, the likelihood of the generative model engaging in plagiarism increases. In the future, this issue should be addressed by developing music-specialized tools for detecting plagiarism.

References

[1] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, 2022.

[2] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

[3] Yi-Jen Shih, Shih-Lun Wu, Frank Zalkow, Meinard Muller, and Yi-Hsuan Yang. Theme transformer: Symbolic music generation with theme-conditioned transformer. *IEEE Transactions on Multimedia*, 2022.

[4] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[5] Christine McLeavey Payne. Musenet. *OpenAI*, April 2019. URL `openai.com/blog/musenet`.

[6] Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann. Figaro: Generating symbolic music with fine-grained artistic control. *arXiv preprint arXiv:2201.10936*, 2022.

[7] Hyun Lee, Taehyun Kim, Hyolim Kang, Minjoo Ki, Hyeonchan Hwang, Sharang Han, Seon Joo Kim, et al. Commu: Dataset for combinatorial music generation. *Advances in Neural Information Processing Systems*, 35:39103–39114, 2022.

[8] Colin Raffel. *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. Columbia University, 2016.

[9] Jeffrey Ens and Philippe Pasquier. Building the metamidi dataset: Linking symbolic and audio musical data. In *ISMIR*, pages 182–188, 2021.

[10] Bee Suan Ong and Sebastian Streich. Music loop extraction from digital audio signals. In *2008 IEEE International Conference on Multimedia and Expo*, pages 681–684. IEEE, 2008.

[11] Nathan Fradet, Jean-Pierre Briot, Fabien Chhel, Amal El Fallah Seghrouchni, and Nicolas Gutowski. Byte pair encoding for symbolic music. *arXiv preprint arXiv:2301.11975*, 2023.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[13] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pages 7176–7185. PMLR, 2020.

Table 1: The evaluation table of loop generation conditioned on musical metadata

Conditions	Dropped	Perplexity (↓)	Density (↑)	Coverage (↑)	Controllability				
					I (↑)	MP (↓)	MT (↓)	MV (↓)	MD (↓)
Full	X	1.589	0.453	0.494	0.969	1.692	2.094e-3	2.763	1.609
Full	O	1.639	0.459	0.484	0.918	2.793	6.881e-3	4.294	2.462
Inst	X	1.935	0.302	0.255	0.686	-	-	-	-
Inst	O	1.607	0.475	0.403	0.899	-	-	-	-

A The Details of Evaluation

We evaluate the capability of generative models and the quality of generated samples in terms of perplexity, density and coverage [13], and controllability. For density and converge, which measure the overlapped ratio between the training and generated set, we project the samples into a latent space using our customized BERT. They each indicate sample fidelity and diversity. For controllability, we report the musical alignment between input conditions and generated samples (the Jaccard index for instruments (I), the absolute difference for mean pitch (MP), mean tempo (MT), mean velocity (MV), and mean duration (MD)).

In Table 1, when full musical conditions are given, the model's performance with the random drops slightly degrades in terms of perplexity, coverage, and controllability. However, when partial musical conditions (only instruments) are given, applying the random drops is noticeably effective since the model has encountered only instrument conditions during training. Thus, it can provide flexible controls to users.