Improvise+=Chain: Listening to the Ensemble Improvisation of an Autoregressive Generative Model

Atsuya Kobayashi [*][†]
Keio University
Endo 5322, Fujisawa City
Kanagawa, Japan
atsuya@sfc.keio.ac.jp

Ryo Nishikado [*][†]
Keio University
Endo 5322, Fujisawa City
Kanagawa, Japan
ryoni118@sfc.keio.ac.jp

Nao Tokui [*][†]
Keio University
Endo 5322, Fujisawa City
Kanagawa, Japan
tokui@sfc.keio.ac.jp

ABSTRACT

This paper describes *Improvise+=Chain*[1], an audiovisual installation artwork of autonomous musical performance using artificial intelligence technology. The work is designed to provide the audience with an experience exploring the differences between human and AI-based virtual musicians. Using a transformer decoder, we developed a four-track (melody, bass, chords and accompaniment, and drums) symbolic music generation model. The model generates each track in real-time to create an endless chain of phrases, and 3D visuals and LED lights represent the attention information between four tracks, i.e., four virtual musicians, calculated within the model. This work aims to highlight the differences for viewers to consider between humans and artificial intelligence in music jams by visualizing the only information virtual musicians can communicate with while humans interact in multiple modals during the performance.

Author Keywords

Symbolic Music Generation, Multi-track Music Generation, Audio-Visual Installation, Visualization

CCS Concepts

•**Applied computing** → **Sound and music computing;** Performing arts; •**Computing methodologies** → *Artificial intelligence;* Knowledge representation and reasoning;

1. INTRODUCTION

1.1 Background

Multi-track symbolic music generation techniques using deep learning have made remarkable progress in recent years. In particular, token-based generation methods using Transformers excel in their expressiveness and the lengths they can handle[1][2][3][4]. Also, several researches focus on the musical interaction during improvisation when developing musical generative system[5][6][7], and various tools or services that apply music generation models have been developed to bring out the expressive capabilities of musicians and to give non-musicians opportunities for musical expression or instant composing.

As for installation with deep music generative models, there are several works. *SONIC PENDULUM* is an installation of 30 pendulums and calming ambient melody generated by Auto Encoder and controlled by CNN model[8]. *Meandering River*[9] is an audiovisual installation comprised of real-time visuals generated by an algorithm and music composed by Performance RNN provided by Google Magenta. Also, *Furniture Music*[10] attempted to reinterpret the music of Erik Satie using a generative music model. However, few works utilize generative models for multi-track music in real-time.

1.2 Concept

An improvised performance by human musicians consists of not only the playing of individual instruments but also complex multi-modal communication such as facial expressions, eye contact, and body movement. It sometimes makes the atmosphere of the ensemble vital, organic, and uncertain. Meanwhile, creative artificial intelligence systems such as music generation models use quasi-randomness to gain variation in their outputs, and there is no inherent uncertainty. The difference should be more evident in improvisation since interactions among multiple creators form dynamic uncertainty. Also, unlike human musicians, the music generation model cannot sense spatial or temporal information for communication. We focused on this determinism and this limitation in communication as the most significant differences between human creative acts and the imitation of human creations via large amounts of data.

This work, *Improvise+=Chain*, is a real-time performance of generative music with piano, guitar, bass, and drums generated by a deep symbolic model. Each part perpetually creates new phrases one after another in jazz improvisation and rock or funk jam, and each is constantly attentive to the other parts' performance, exchanging information and influencing each other. We expressed this multidirectional interaction through computer graphics animation and the streaming lines of light connecting the speakers in this installation. We modeled musicians' instrumental performances and interactions and aimed for the viewer to explore through experience how their behavior differs from that of human musicians and what musical value can be found in it.

[*] Graduate School of Media and Governance
Keio University Shonan Fujisawa Campus

[†] Computational Creativity Lab.
`https://cclab.sfc.keio.ac.jp`

[1] Performance demo videos, Digest: `https://youtu.be/wiFhfswgsMU` Full: `https://youtu.be/kfiWtT3ZQgw`

NIME'23, 31 May–2 June, 2023, Mexico City, Mexico.

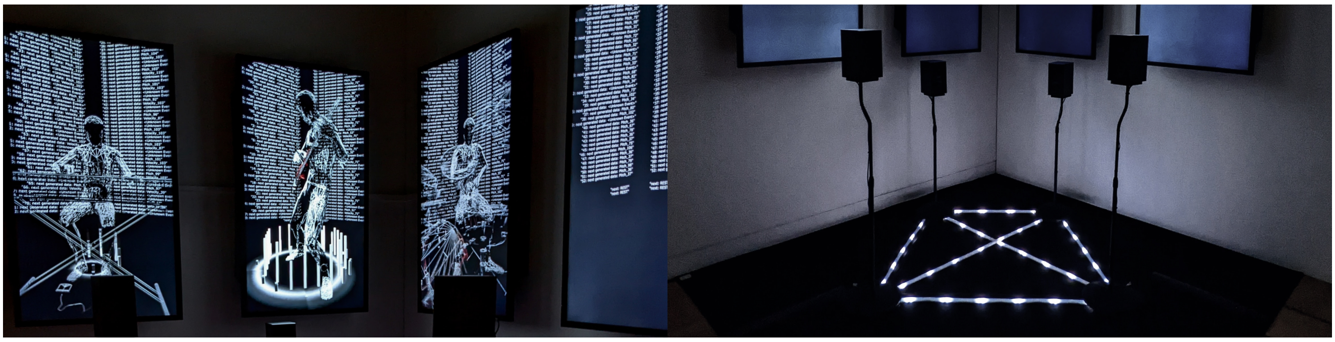

Figure 1: A (left): 4 displays are installed on the wall, each showing virtual musician's performance by visualizing audio wave, animation synced to the tempo, and program console outputs. B (right): 6 LED tapes are installed on the floor connecting each pair of 4 speakers, which illustrates the amount of reference information on the generation of each part.

2. SYSTEM DESIGN

2.1 Data Preparation and Model Training

We trained a causal Transformer Decoder model (embedding dimension is 128, context length is 2048, 8 layers, 8 attention heads for each layer, batch size is 8) using Huggingface Transformers[2], and the dataset is ≈ 1500 MIDI files extracted from the Meta MIDI Dataset[11]. We selected songs with the time signature of 4/4, and were tagged `Jazz`. After converting each song to one with only four tracks with Midi-Miner[12], we cut out 8-bar length chunks, sliding 4-bar each from each piece of song. To create a token dataset for the model training, we tokenized each MIDI datum of chunk with REMI encoding[13] with additional `<Track>` and `<Instrument>` tokens used in MMM-Track model[1] (Appendix A) (vocabulary size is 388), to support the autoregressive track addition task. This tokenizer is implemented with MidiTok library[14]. The training was performed for 5 epochs using an NVIDIA RTX3090 (80% for training data, learning rate is `1e-4` with linear scheduler, the optimizer is Adam).

2.2 Music Generation and Sound Design

The main system flow is chaining the track generation in real-time. We designed a generation server and player client application to work in sync. The generation server is written in Python 3.8 and based on a single OSC[3] server process, which handles triggers of new track generation on every 8 bars from the client application. This process has an internal state which can control generation sampling parameters and tempo. As a client application, we used Ableton Live 11 Suite[4] to construct a real-time variable tempo performance system. Generated MIDI sequences are played back using the sampler plugins[5]. Four powered speakers[6] are driven from an audio interface with 8 channels audio outputs[7].

2.3 3D Graphic and Light Visualization

The behavior of the virtual performers is represented using 3D models collected from several online services[15][16][17][18], and the scene created in TouchDesigner[8] consists of four 3D models with bone animations of instrumental performances got from Mixamo[9], moving at synchronized tempo to the ensemble. The waveform underfoot is audio-reactive, visualizing each part as it plays. The background of each model is the log output at the time of generation, representing what is going on in their heads (Figure 2).

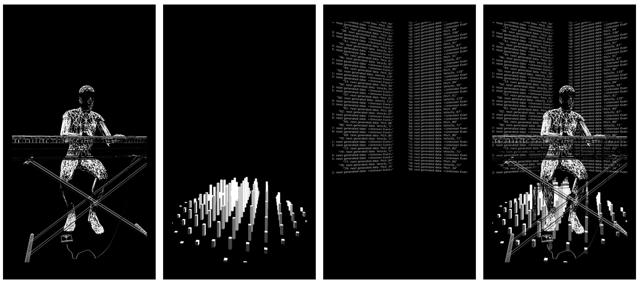

Figure 2: A 3D Graphic model and visualized layers: 3D model with bone animation, audio gain visualization, and console outputs from the server.

At each track inference, the system obtains matrix of Attention weights at each time step as well as generated tokens, and calculates how much attention the generated tracks give to other tracks; The index of the first token of each source track ($i \in [1...4]$) in the priming sequence is $P_i < 2048$ and the length of generated token sequences for each target track ($j \in [1...4]$) is $G_j < (2048 - P_4)$. The number of heads and layers are $M > 0$ and $N > 0$, respectively. Then, the score matrix $Score_{i,j}$ is computed from the attention tensor $Attn^{(j)}$ resulting from generating target track j as follows

$$Score_{i,j} = \frac{1}{G} \sum_{g=1}^{G_j} \max_{\substack{n=1...N,\, m=1...M \\ p=P_i...P_{i+1}-1}} Attn^{(j)}_{g,p,n,m}$$

The calculated $Score$ matrix is visualized on six LED tapes, which chains WS2812 LED is laid between the speakers (Figure 1:B) and also displayed on the user interface for installation (Figure 3). These tapes are controlled by the sketch written with Adafruit NeoPixel[10], running on an Arduino Mega networked to the generation server. The attention information is sent in real-time to the Arduino, which

[2] `https://huggingface.co/docs/transformers/v4.25.1/en/model_doc/gpt2#transformers.GPT2LMHeadModel`

[3] `https://opensoundcontrol.stanford.edu/`

[4] `https://www.ableton.com/en/live/`

[5] We used Jazz Guitar, Electric Piano, and Classic Bass included in KONTAKT 7 Player and Session Kit Ableton Pack `https://www.native-instruments.com/en/products/komplete/samplers/kontakt-7-player/`, `https://www.ableton.com/en/packs/session-drums/`

[6] `https://www.fostex.jp/products/pm03/`

[7] `https://focusrite.com/en/usb-audio-interface/scarlett/scarlett-18i8`

[8] TouchDesigner 2022.31030 `https://derivative.ca/`

[9] `https://www.mixamo.com`

[10] `https://github.com/adafruit/Adafruit_NeoPixel`

changes the direction and speed of the LED light stream in 10 steps. Light flows from the source track to the target track.

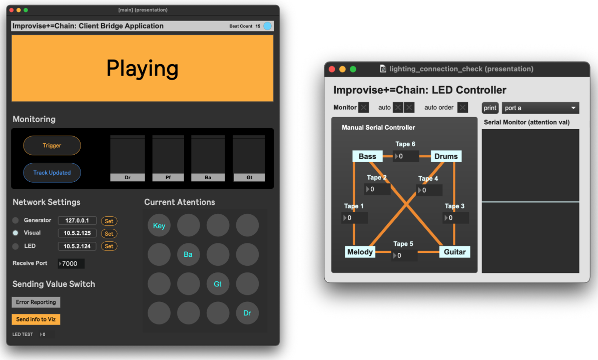

Figure 3: User Interfaces for control, A (left): Max for Live device window for operating music player and monitoring the generation server. B (right): Max patcher for controlling, monitoring, and testing the LED controller device (Arduino).

3. EXPERIENCE DESIGN

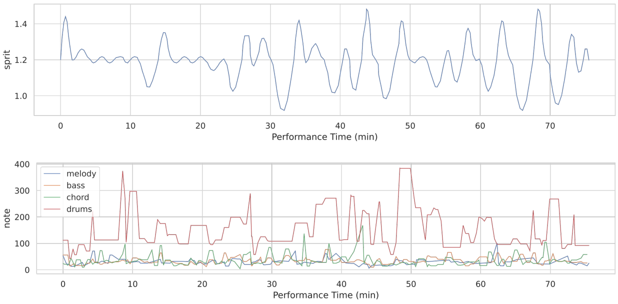

Figure 4: A (above): Time series log of internal *Energy* value and B (below): the number of generated notes for each part, in an 80 minutes performance.

This work aims to give the audience a sense of how an improvised ensemble transitions through the performance and how it differs from humans. To this end, we have designed several mechanisms that mimic the improvisational ensembles of human musicians.

3.1 Internal Algorithms

Re-generation is triggered every 8 bars, selecting one or more of the tracks currently being played and then replacing the parts with a newly generated sequence. The combination of tracks to be regenerated is selected randomly from the predetermined pattern list (Appendix B). Tempo information is not handled by the generation model, but is maintained by the generation server as an internal state called *Energy* that represents the dynamics of the ensemble using one-dimensional Perlin noise with a period of eight steps (Figure 4:A). The initial tempo is determined by a random number sampled from $Uniform(60, 160)$. At the initial generation, all four tracks are generated at once with the parameters for the method of sampling from output Softmax: $Temperature \in [0.9, 1.5)$ randomly sampled from the Uniform distribution of the range. The $Energy^i \in [0, 1.0]$ is updated every i-th 8-bar play, and it determines the range of tempo change ($Tempo^i = min(max(Energy^i \times 100 + 60, 40), 190)$), and parameters for generation these affect the number of notes generated ($Temperature^i = Energy^i \times 0.6 + 0.9$) (Figure 4:B). In addition, an overall initialization (sampling of tempo, $Temperature$ and generation of all four tracks) is re-performed every 128 bars, causing abrupt changes in the performance.

3.2 Installation

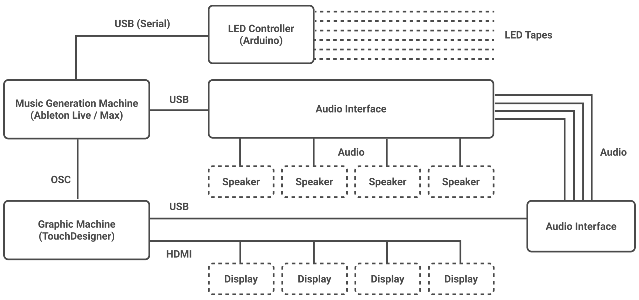

Figure 5: System Diagram

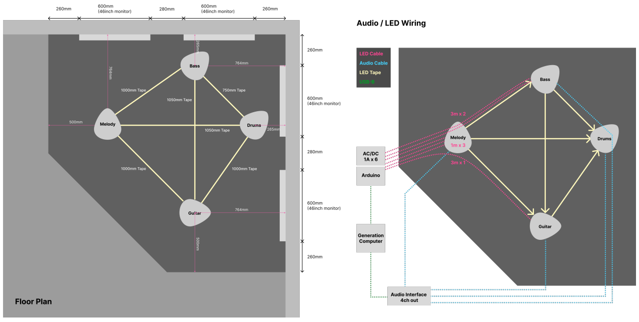

Figure 6: Floor Plans

At least two machines with a middle-class graphic processor connected to the same network are needed to perform this installation work (Figure. 5). The work should be performed in a 2-meter square area in a quiet corner of the room, with four displays, four speaker stands, and 6 LED tapes (Figure.6).

4. CONCLUSION AND FUTURE WORK

We designed an audio-visual experience of listening to the ever-changing ensembles and implemented the audio-visual installation work of performance by four virtual musicians using a deep neural network model for multi-track symbolic music generation in real-time. Also, to emphasize the difference between machines and humans, the work visualizes the model's internal state as communication between virtual musicians. In the future, we plan to improve the variety of musical expressions of the model and the audio-visual system to achieve a more realistic and lifelike performance.

5. ETHICAL STANDARDS

During the exhibition, we took care not to cause any health hazard due to the loud volume. This research does not include studies with human participants. No animals were involved.

6. REFERENCES

[1] Jeff Ens and Philippe Pasquier. Mmm : Exploring conditional multi-track music generation with the transformer, 2020.

[2] Cong Jin, Tao Wang, Xiaobing Li, Chu Jie Jiessie Tie, Yun Tie, Shan Liu, Ming Yan, Yongzhi Li, Junxian Wang, and Shenze Huang. A transformer generative adversarial network for multi-track music generation. *CAAI Trans. Intell. Technol.*, 7(3):369–380, September 2022.

[3] Hao-Wen Dong, Ke Chen, Shlomo Dubnov, Julian McAuley, and Taylor Berg-Kirkpatrick. Multitrack music transformer. July 2022.

[4] Cong Jin, Tao Wang, Shouxun Liu, Yun Tie, Jianguang Li, Xiaobing Li, and Simon Lui. A Transformer-Based model for Multi-Track music generation. *IJMDEM*, 11(3):36–54, July 2020.

[5] S H Hakimi, N Bhonker, and R El-Yaniv. Bebopnet: Deep neural models for personalized jazz improvisations. *program.ismir2020.net*, 2020.

[6] Vincenzo Madaghiele, Pasquale Lisena, and Raphael Troncy. MINGUS: Melodic Improvisation Neural Generator Using Seq2Seq. In *22^{nd} International Society for Music Information Retrieval Conference (ISMIR)*, Online, 11 2021.

[7] Olga Vechtomova and Gaurav Sahu. LyricJam sonic: A generative system for Real-Time composition and musical improvisation. October 2022.

[8] Yuri Suzuki. Yuri suzuki "SONIC PENDULUM" sound installation using AI to generate music from environmental sounds. `https://qosmo.jp/en/projects/sonic-pendulum/`, April 2017. Accessed: 2023-1-20.

[9] Cedric Kiefer. Meandering river audiovisual art installation. `https://onformative.com/work/meandering-river/`, 2018. Accessed: 2023-1-20.

[10] Nao Tokui. Furniture music(s) 2021—A multi-channel AI generative music installation. `https://naotokui.net/en/works/furniture-musics-2021/`, October 2021. Accessed: 2022-12-27.

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

[12] Rui Guo, Dorien Herremans, and Thor Magnusson. Midi miner - A python library for tonal tension and track classification. *CoRR*, abs/1910.02049, 2019.

[13] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[14] Fradet Nathan, Briot Jean-Pierre, Chhel Fabien, El Amal, Seghrouchni Fallah, and Gutowski Nicolas. Miditok: A python package for midi file tokenization. In *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference*, 2021.

[15] Guitar 3D model. `https://www.highend3d.com/3d-model/guitar-3d-model-79058`. Accessed: 2023-1-19.

[16] Bass guitar low poly freebie - download free 3D model by geug, July 2019.

[17] Submitted by Dgemmell. Drum set free 3D model - .blend .obj - Free3D. `https://free3d.com/3d-model/drum-set-41081.html`. Accessed: 2023-1-19.

[18] Mike. Yamaha piano set. `https://3dwarehouse.sketchup.com/model/9672ee3dc322de0ac4a615a7bc178f5a/Yamaha-Piano-Set?hl=en&login=true`. Accessed: 2023-1-19.

APPENDIX

A. TOKEN REPRESENTATION

```
<Track Start>
    <Instrument value="Melody">
    <Bar Start>
      <Pitch value=64>
      <Position value=1.0.0>
      <Velocity value=80>
      <Duration value=0.0.4>
        :
    <Bar End>
      :
<Track End>
<Track Start>
    <Instrument value="Bass">
     <Bar Start>
  :
```

B. RE-GENERATION PATTERN LIST

Priming	**+ Generate**	**- Remove**
Mel. Ba. Cho. Dr.	Mel.	Ba. Cho.
Mel. Cho. Dr.	Ba.	Mel. Cho.
Mel. Ba. Cho.	Dr.	Mel. Ba. Cho.
Mel.	Ba. Cho. Dr.	–
Mel.	Cho. Dr.	–
Mel. Dr.	Ba. Cho.	–
Mel. Dr.	Cho.	–
Ba.	Cho.	–
Ba.	Cho. Dr.	–
Cho.	Ba. Dr.	–
Cho.	Mel. Ba. Dr.	–
Dr.	Ba.	–
Dr.	Ba. Cho.	–
Dr.	Mel. Ba.	–
Ba. Dr.	Cho.	–
Ba. Dr.	Mel.	–
Cho. Dr.	Ba.	–
Cho. Dr.	Ba. Mel.	–
Ba. Cho.	Dr.	–
Ba. Cho.	Mel. Dr.	–
Ba. Cho. Dr.	Mel.	–
Ba. Cho. Dr.	Ba. Cho.	–
Ba. Cho. Dr.	Dr.	–
Mel. Ba. Cho. Dr.	Ba. Cho.	Mel.
Mel. Ba. Cho. Dr.	Dr. Cho. Ba.	–
Mel. Ba. Cho. Dr.	Dr. Cho. Ba.	Mel.
Mel. Ba. Cho. Dr.	Ba.	Mel. Cho.
Mel. Ba. Cho. Dr.	Ba. Dr.	Mel. Cho.
Mel. Ba. Cho. Dr.	Mel.	–
Mel. Ba. Cho. Dr.	Mel. Cho.	–
Mel. Cho. Dr.	Ba.	–
Mel. Ba. Dr.	Cho.	–
Mel. Ba. Cho.	Dr.	–
Ba. Cho. Dr.	Mel.	–