Scramble Live: Combining LSTM and Markov Chains for Real-time Musical Interaction.

Nicola Privato
Conservatorio C.Pollini of Padua
nicola.privato@gmail.com

Omar Rampado
omar@ognibit.it

Alberto Novello
Conservatorio C.Pollini of Padua
alberto.novello@conservatoriopollini.it

ABSTRACT

Scramble Live is a complementary tool for *Scramble*[1], a standalone MIDI software developed in Max/MSP[2] for the real-time generation of polyphonic music [1]. At their core, both tools combine the flexibility of Markov Chains for the generation of pitch transitions with the generalization capabilities offered by LSTM neural networks on the rhythmic and dynamic domains. *Scramble* performs the analysis of MIDI files, allows for the selection of the LSTM hyperparameters and the training of the neural network through an intuitive user interface, but offers limited real-time interactivity on playback. *Scramble Live* relies on the models generated by *Scramble*, but allows for a high degree of interaction with live instrumentalists and with the system itself. To complete its performance-focused capabilities, *Scramble Live* offers MIDI mapping and synchronization to external clock sources.

INTRODUCTION

Markov Chains (MC) and Recurrent Neural Networks (RNNs) are two of the most common approaches to algorithmic composition by the means of computer-based applications [2]. Each of the two has specific advantages and limitations: RNNs can be very consistent in their predictions and perform well on generalization tasks. On the other hand they require long training time and large datasets, and do not offer much flexibility once they are trained. MCs provide a greater degree of real-time control over the produced output [3] but are inherently incapable of generalizing [4]. Additionally, the more musical dimensions are incorporated in each state (e.g. pitch and velocity), the less likely transitions become.

Scramble and *Scramble Live* take advantage of the specificities of both approaches. By combining a MC for the generation of pitch transitions with a *Long Short-Term Memory* (LSTM) RNN [5] for all time-related and dynamic variables, the real-time interaction is preserved, and the neural network's training time is reduced. Furthermore, it is possible to include in the system the live input from any MIDI instrument while offering satisfactory generalization capabilities and a high degree of rhythmic and expressive variability depending on the selected LSTM model.

We chose to develop both *Scramble*'s iterations in Max/MSP because, even though the platform offers some machine learning libraries [6], the application of more complex algorithms such as LSTM RNNs remains for the most part unexplored.

1. SCRAMBLE LIVE

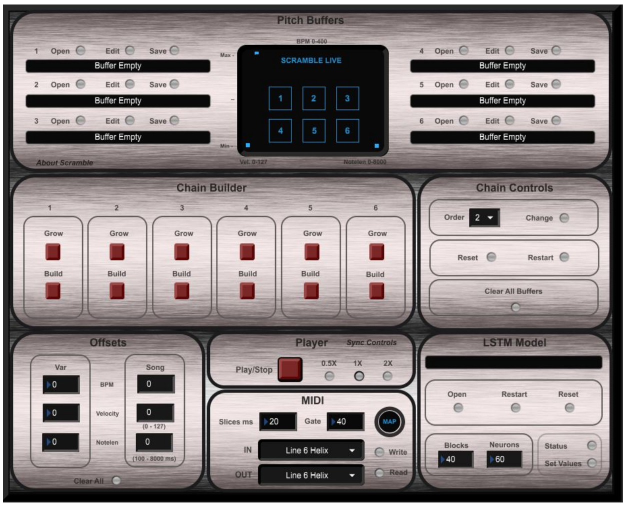

Figure 1. *Scramble Live*'s GUI.

Scramble Live is a complementary tool for *Scramble*. It is focused on live use, offering the musician an intuitive *Graphical User Interface* (GUI) that can be easily navigated while performing. In order to make the user's experience as intuitive as possible, we did not include into *Scramble Live*'s GUI the MIDI player and LSTM training sections. All LSTM models providing rhythm, tempo and velocity data to the pitch transitions generated by the MCs, must be therefore trained on and stored from *Scramble*, and subsequently opened in *Scramble Live.*

The tool offers six buffers where it is possible to temporarily store, save and load the pitch information of any MIDI sequence analyzed and saved from the main software's MIDI player, or played live from an instrument connected to the MIDI input. The data of any number of buffers, in any desired order, can be manually edited and used to generate a MC transition table. It is

[1] https://www.dropbox.com/sh/v869gify6pm6ta7/AAAcEjlWj-kIJ2VpyNfCWHN7a?dl=0
[2] https://cycling74.com/products/max

possible, for instance, to combine the pitches of a Bach invention with the real-time input of a pianist, coherently paired by the LSTM neural network with the rhythm, tempo and velocity of a model trained on Scott Joplin's ragtime style.

Furthermore, *Scramble Live* offers the possibility to change the MC transition table in real time without compromising the rhythmic structure, thus allowing the user to continuously interact with the tool and/or with any musician connected to the MIDI input.

Common to both iterations, and particularly effective in changing the MIDI output to the user's need, is the *Offsets* menu, which allows to add a positive or negative offset to the BPM, note lengths and velocity values generated by the LSTM, or to change playback to half or double time. In addition, *Scramble Live* offers the possibility to bypass the LSTM BPM variations and connect to an external master clock, in order to easily integrate the tool with any given electroacoustic setup.

In order to allow for the use of *Scramble Live* through hardware MIDI controllers, an automatic mapping function has been added. The pairing is performed by activating the mapping, selecting a playback, MC or buffer control on the GUI, and pressing the desired hardware switch. The mapping configuration can then be stored and opened at will.

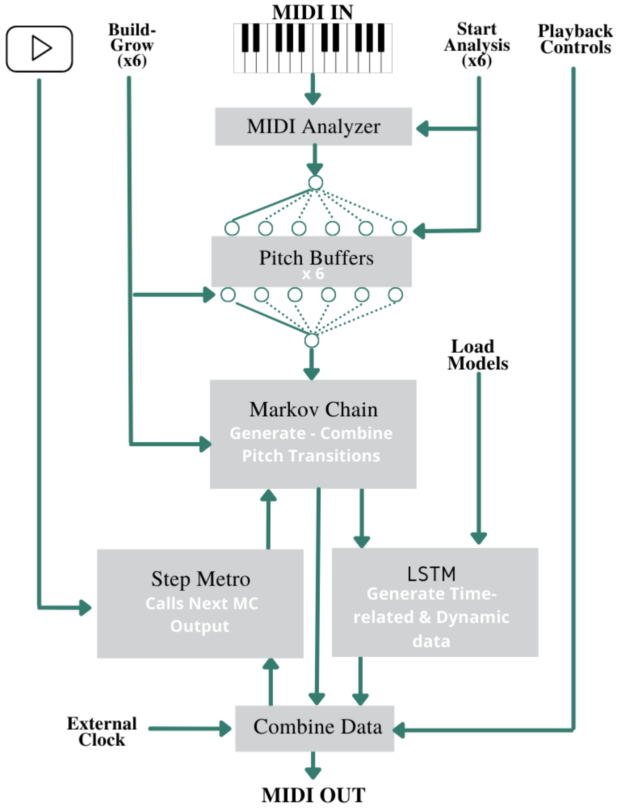

Figure 2. *Scramble Live*'s block diagram.

2. FUTURE DEVELOPMENTS

We presented *Scramble Live*, a tool for the real-time dynamic generation of polyphonic music, that extends the capabilities of *Scramble* to the performative scope. The system is based on two Max/MSP external modules: an obsolete LSTM developed by Wesley Jackson (2011)[3] and a MC from the *ml.star* library by Benjamin Day Smith [7]. The LSTM module initially presented some major bugs and design limitations that have been fixed or worked around, such as compatibility with Max 8, import and export methods, separate threading implementation and introduction of batch load for all training data. We are currently working on a new LSTM external that would allow for higher performances and architectural flexibility. The new LSTM iteration will be integrated in *Scramble* and *Scramble Live*, and become the core of a Max/MSP RNN package.

1. REFERENCES

[1] N. Privato, O. Rampado, and A. Novello, "A creative tool for the musician combining LSTM and Markov Chains in Max/MSP" in Proc. Int. Conf. on Artificial intelligence in Music, Sound, Art and Design (EvoMUSART2022), Madrid, 2022, (to be published after the conference -April 22nd 2022).

[2] H. Kumar and B. Ravindran, Polyphonic music composition with lstm neural networks and reinforcement learning. arXiv preprint arXiv: 1902.01973, 2019.

[3] A. Cleermans, D. Servan-Schreiber, and J.L. McClelland, Finite State automata and Simple Recurrent Networks. Neural Computation 1(3), 372-381(091989). https://doi.org/10.1162/neco.1989.1.3.372

[4] A. Papadopoulos, P. Roy, and F. Pachet, "Avoiding plagiarism in markov sequence generation" in Proceedings of the AAAI Conference on Artificial Intelligence. vol.28, 2022.

[5] S. Hochreiter, J. Schmidhuber, and A.S. McGough, Long-short term memory. Neural computation 9(8), 1735-1780, 1997.

[6] I. Simon, A. Roberts, C. Raffael, J. Engel, C. Hawthorne and D. Eck, Learning a latent space of multitrack measures. arXiv preprint arXiv: 1806.00195v1, 2018.

[7] B.D Smith and G.E. Garnett, "Unsupervised Play: Machine learning toolkit for max". NIME 2012.

[3] From J. Franklin's source code (2004), based on F. Gers, N. Schraudolph and J. Schmidhuber pseudocode (2002)