AN LSTM-BASED DYNAMIC CHORD PROGRESSION GENERATION SYSTEM FOR INTERACTIVE MUSIC PERFORMANCE

Christos Garoufis[1,2]*, Athanasia Zlatintsi*[1,2]*, and Petros Maragos*[1,2]

[1]School of ECE, National Technical University of Athens, Zografou 15773, Greece
[2]Robot Perception and Interaction Unit, Athena Research Center, 15125 Maroussi, Greece
cgaroufis@mail.ntua.gr; [nzlat, maragos]@cs.ntua.gr

ABSTRACT

In this paper, we describe an interactive generative music system, designed to handle polyphonic guitar music. We formulate the problem of chord progression generation as a prediction problem. Thus, we propose utilization of an LSTM-based network architecture incorporating neural attention that is able to learn a mapping between symbolic representations of polyphonic chord progressions and future chord candidates. Furthermore, we have developed a virtual air-guitar controller, utilizing a Kinect device, that uses the above architecture in order to change in real time the guitar chord mapping, depending on the performer's previous performance. The whole system was evaluated both objectively and subjectively. The goal of the objective evaluation was to measure the ability of the system to correctly generate chord candidates for existing chord progressions, as well as identify the type of errors. The subjective evaluation mainly focused on the longer-term behavior of the system, regarding the musical coherence and the variety of the generated progressions. The results were encouraging regarding the ability of our system to generate sound chord progressions, while highlighting a number of issues that require to be resolved.

Index Terms— chord progression generation, human- computer interaction, LSTM networks, interactive performance system, chord prediction

1. INTRODUCTION

In recent years, the rapid progress in the fields of artificial intelligence and machine learning has led to an influx of development in automatic music generation. While earlier attempts in the field leaned on the use of Hidden Markov Models (HMMs) [1] in order to capture the temporal dependencies in sound evolution, various deep network architectures, such as Recurrent Neural Networks (RNNs) [2, 3], Generative Adversarial Networks (GANs) [4] and autoencoder architectures [5] have been more recently utilized in order to create music generation systems. These systems are able, for instance, to handle polyphony [4] and learn to synthesize music using composer-specific characteristics [6]. In specific, LSTM-based autoencoders have found use in a plethora of applications, further including polyphonic music generation [7] and learning long-term temporal dependencies in music [8].

Despite the above, one area that still has lots of unrealized potential is the integration of interaction in computer music generation. This interaction can take a number of forms, usually including the ability to interfere with a number of musical parameters [6], which can aid in collaborative music synthesis between a human performer and a generative AI. Furthermore, the appearance of non-intrusive motion sensors such as Microsoft Kinect or Leap Motion have made easier the development of advanced multimodal interfaces, through which a performer can interact with virtual music instruments that can be either simulations, or augmentations, of real-world instruments [9] or abstract explorations of musical spaces [10].

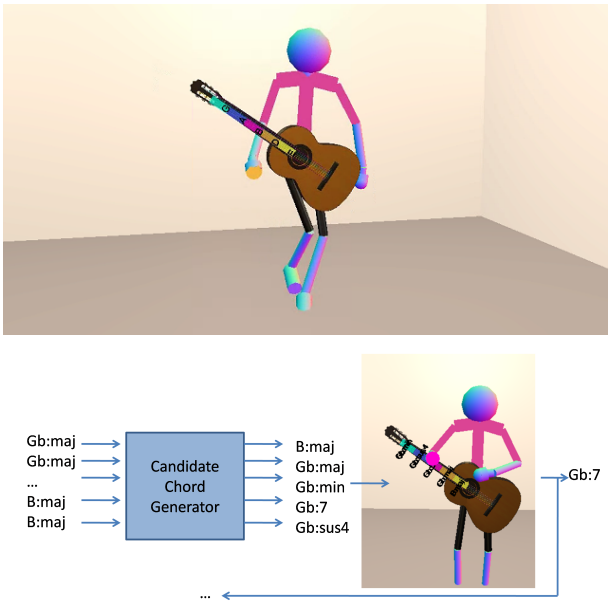

Fig. 1. Top: Snapshot of the developed environment for interacting with a virtual air guitar, Bottom: Example of interactive chord progression generation using our system.

The work in this paper lies in the intersection between interactive and generative music systems, suited for interactive polyphonic guitar music generation. We formulate the generation of chord progressions from symbolic representations as a prediction problem. As such, we experiment with LSTM-based neural networks, mapping chord sequences to their most probable following chords. In the same vein, we incorporate neural attention mechanisms [11] in our system, investigating their effect in the overall performance.

Furthermore, the chord progression generator has been incorporated into a multimodal interface [12] that facilitates interaction between a performer and a virtual air-guitar, using a Kinect device (see Fig. 1 Top). Depending on specific gestures from the performer's part, a different candidate chord is triggered, according to which the set of candidate chords is updated in real time (see Fig. 1 Bottom). In this context, this work could be considered an example of "Creative MIR" [13].

Proceedings of the 45th International Conference on Acoustics, Speech and Signal Processing (ICASSP-2020), Barcelona, Spain, May 2020

The rest of the paper is organized as follows: A literature review in interactive generative music systems is conducted in Sec. 2. In Sec. 3, we present the architectures we experimented with, addressing the various design choices, and describe the process we followed with regards to the dataset. An objective evaluation of the chord progression generator is carried out and described in Sec. 4, while in Sec. 5 we present a subjective evaluation of the interactive system as a whole. Finally, conclusions and some possible future work directions are described in Sec. 6.

2. RELATED WORK

Interactive and generative music systems can be combined in two ways – either by introducing the ability to alter some implicit [6, 14] or explicit [15] musical features on otherwise automatic generative systems, or by the addition of generative elements in interactive setups and installations. An example of such a strategy is described in [16], where, following an initial mapping between control vectors and musical parameters provided by the performer, the system learns to generate music that is compatible with any user gesture that corresponds to any potential control vector. Jive [17] is a generative music system based on an evolutionary algorithm, where the user can interact with it by providing scores to the system's musical output, thus helping the output to evolve towards their preferences.

A number of works have been published in recent years by Magenta in the field of music generation, including Attention RNN [18], which is trained using one-hot encoded sequences of musical events. Similarly, Piano Genie [5] is an interactive piano controller, trained as an end-to-end encoder-decoder with monophonic piano sequences, and possessing a discretized intermediate latent layer.

Another work that lies in the boundary of polyphonic music generation and interactivity – in the form of co-improvisation between a human performer and an AI – is that of Dong et al. [4, 19]. The main idea of this work is that, given a complete melody sheet for a specific band instrument, melody sheets for the rest of the instruments can be generated. In [4], Generative Adversarial Networks (GANs) were utilized in order to create the melodic sheets in the form of binary pianorolls, while in an extension of this work in [19], the output layer was augmented with the use of binary neurons.

Finally, a number of research works have been recently carried out regarding guitar tablature generation. For instance, [20] reports in the development of an intelligent tablature creator, using a genetic algorithm in order to find the optimal hand positioning for each chord with regards to minimizing the hand movement, while [21] employs a probabilistic model that generates guitar solo tablatures from pre-defined chords and keys.

3. METHODOLOGY

3.1. Overall Architecture

The overall architecture of our system is presented in Fig. 2. An LSTM-based neural network is employed, in order to generate candidate chords from chord progression sequences. Whenever a chord event is triggered, the input chord progression of the generator is updated, and a new set of candidate chords is calculated.

To facilitate the interaction between a performer and the system, a multimodal interface has been developed, deploying a Kinect sensor. Whenever a performer stands in front of the sensor, a skeletonized avatar appears in the computer screen, including a virtual air guitar, mounted around their waist. The performer can excite the virtual instrument by performing plucking gestures with their dominant

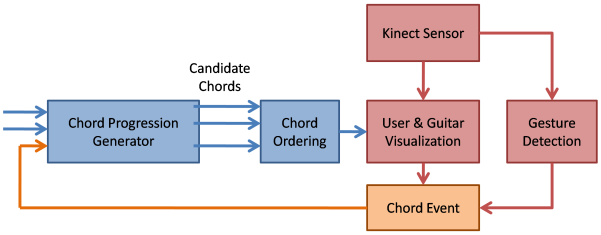

Fig. 2. An overview of the whole system architecture.

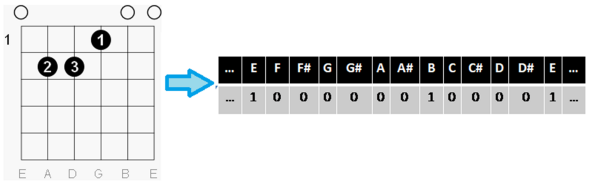

Fig. 3. The E:major guitar chord (left), and its pianoroll notation (right).

hand, moving it vertically at the height of their waist. The proposed candidate chords are placed in the fretboard of the air-guitar, in descending order, and whenever a plucking gesture is recognized, a chord is played, depending on the positioning of the subdominant hand of the performer in the fretboard [12].

3.2. Dataset and Feature Representation

For the training of our system, we used a modification of the Mc-Gill Billboard dataset [22], which contains chord sequences of all weekly no. 1 Billboard hits between 1958–1992, along with a number of other features, such as the dominant instrument in each song.

In order to get a technically correct representation of chord progressions, we initially acquired the chord progression from each song, using the song's temporal scale as the resolution of the chord progression. Next, we reduced the size of the dataset, pruning all songs where the guitar was not the dominant instrument. Finally, after removing duplicates, we were left with a total of 442 songs, or 192,869 chords.

A total of 560 unique chords were present in the dataset, and thus, a simplified chord vocabulary was used, consisting of the following chord types: *maj*, *min*, *7th*, *dim*, *aug*, *sus2*, *sus4*, *ext* (9ths, 11ths, etc), *5th* (power chords) and *1st* (bass notes). All 12 possible chord chromas were considered, thus reducing the vocabulary size to 121 chords, including silence.

To represent the various guitar chords, we use the pianoroll notation, where the input takes the form of an array, where columns correspond to time intervals, and rows to specific notes. The appearance of a specific note in a chord event is denoted by a value of 1 in the corresponding array cell, otherwise, that cell is set to 0. An example pianoroll representation for a chord is portrayed in Fig. 3.

3.3. Candidate Chord Generator Architecture

A block diagram architecture of the LSTM network we utilized in order to generate candidate chords is presented in Fig. 4. We split the problem of candidate chord generation in two: namely, we train an auxiliary network to detect when the chord progression changes to a different chord, and an LSTM-based network using as training data only the chord progression sequences where a chord switch occurs.

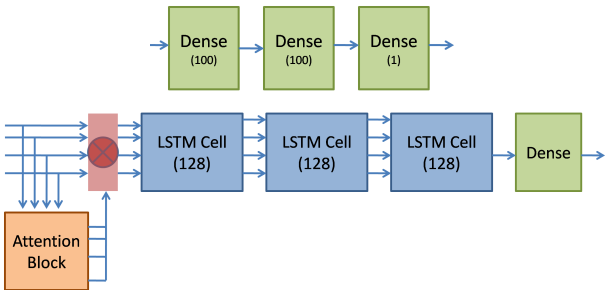

Fig. 4. A layer-level overview of the proposed chord switch detection (top) and candidate chord progression generator (bottom) networks.

The auxiliary network consists of two Dense layers, of 100 neurons each, and a final neuron that predicts on whether a chord switch will occur in the following timestep. For chord progressions of length l, the input of the auxiliary network is a binary vector s, with $l-1$ elements, that equals:

$$s[t] = \begin{cases} 1, C_{t+1} \neq C_t \\ 0, \text{otherwise.} \end{cases} \quad (1)$$

On the other hand, the candidate chord generator consists of a neural attention mechanism, three uni-directional LSTM layers, of 128 units each, as well as a final Dense layer, consisting of neurons equal to the input representation size. The fact that chord progressions in popular music have a somewhat repetitive structure, often repeating the same chords after 8 or 16 quarters, motivated us to include the attention mechanism in our architecture.

It is worth noting that we do not formulate the problem as a classification one, but as a regression one, since the network is trained to generate chord pianorolls. In order to introduce an interactive component in an otherwise generative architecture, we keep as potential outputs of the system a number of guitar chords, based on their Euclidean distance between their representations and the network output. For deploying the above architecture in the developed interface, as a compromise between artistic expressivity and the sensing capabilities of Microsoft Kinect, we set that number as equal to 5. Moreover, when the switch detection mechanism does not predict a chord change, these 5 chords are chosen as the previous chord and the top 4 candidate chords.

However, this architecture has no intrinsic way to produce a meaningful topological mapping of the output chords, since its final output is a set of unordered vectors. In an attempt to solve this problem, we calculate an one-dimensional mapping of the various guitar chords, where each chord is assigned an integer index. To this end, we utilize an evolutionary algorithm, intending to minimize the following loss function, over all possible chords i, j in our corpus C:

$$\sum_{i \in C} \sum_{j \in C} \left(\frac{d\_sonic(i,j)}{d\_spatial(i,j)} \right), \quad (2)$$

where d_sonic corresponds to the Euclidean distance between two chord representations and $d\_spatial$ is the distance of the two chords in the mapped space. In practice, the system orders the predicted chords using the mapping that minimizes the above quantity.

4. OBJECTIVE EVALUATION AND DISCUSSION

The dataset was split into training, validation, and testing data, in a 3:1:1 ratio, and the data were standardized before being fed to the network. The proposed architecture (denoted as A+S) was evaluated against a baseline model that includes neither the switch predictor nor the attention module (B), an architecture that includes the switch predictor but not the attention module (S) and a model where the attention module is connected directly to the latent space before the last LSTM layer (SA+S). All architectures were trained for 20 epochs, using 5-fold cross validation, the Adam [23] optimizer with a learning rate of 0.001, and batch size equal to 128. We used both top-1 and top-5 chord prediction accuracy as metrics, since the interactive component of our system is designed to handle 5 chords simultaneously.

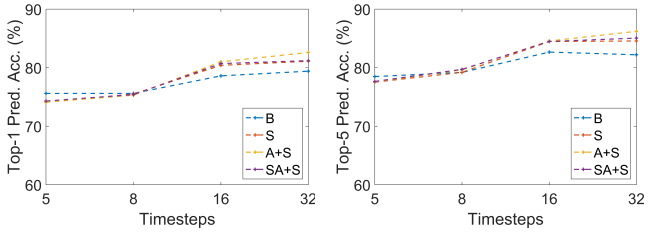

Fig. 5. Top-1 (left) and top-5 (right) chord prediction accuracy for all of the tested architectures, with respect to the chord progression length (in timesteps).

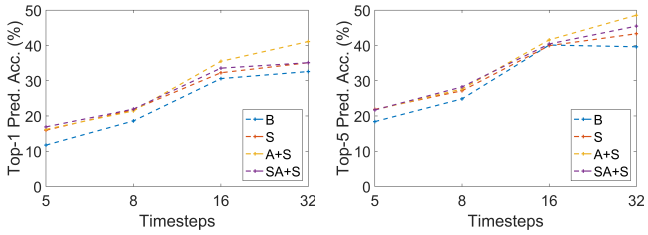

Fig. 6. Top-1 (left) and top-5 (right) chord prediction accuracy in the cases where a chord switch occurs, for all of the tested architectures, with respect to the chord progression length (in timesteps).

The results of this comparative analysis, for various input chord progression lengths (in timesteps), are shown in Figures 5, 6. For small sizes of the input chord progression, we can see that neither the switch detection mechanism nor the attention module affect positively the overall prediction accuracy. Increasing the input chord progression length to 8 and 16 timesteps yields a more concrete effect in the chord prediction accuracy, that is mainly attributed to the switch detection mechanism. However, while both architectures without a neural attention mechanism seem to plateau at 16 timesteps, the switch-attention network (A+S) achieves higher chord prediction when 32 timesteps are used. The attention mechanism seems to be successful in identifying the most important instances in the chord progression sequence in order to predict the following chords. Finally, we may note that connecting the input attention mechanism to the output of the second LSTM layer yields a behavior similar to the cases where no attention mechanism is utilized.

In Fig. 6, the chord prediction accuracy in the cases where the played chord is different to the one played at the previous timestep is presented. We note that, rather expectedly, it is significantly lower compared to the overall accuracy. In contrast to the overall accuracy, we can see that the baseline architecture does not outperform the modified ones, for any input chord progression length. Finally, as seen in Figs 5, 6, the top-5 accuracy follows the same trends as the top-1 accuracy, with respect to both the input chord progression length and network architecture.

Setup	Top-1 %			Top-5 %		
	Acc.	C.Acc.	T.Acc.	Acc.	C.Acc.	T.Acc.
B	79.40	81.26	85.00	82.20	86.28	93.81
S	81.05	82.74	85.96	84.56	**91.44**	**97.54**
A+S	**82.60**	**84.34**	**86.6**	**86.21**	91.17	97.24

Table 1. Top-1 and top-5 chord prediction accuracies, regarding the chord (Acc.), the chord chroma (C.Acc.) and the chord type (T.Acc), for the baseline (B), switch (S) and attention-switch (A+S) architectures.

Setup	Top-1 %			Top-5 %		
	Acc.	C.Acc.	T.Acc.	Acc.	C.Acc.	T.Acc.
B	32.60	38.66	**56.03**	39.62	52.21	**80.21**
S	35.12	40.66	48.06	43.32	52.22	66.10
A+S	**41.06**	**48.67**	51.01	**48.58**	**58.32**	66.81

Table 2. Top-1 and top-5 chord prediction accuracies, regarding the chord (Acc.), the chord chroma (C.Acc.) and the chord type (T.Acc), for the baseline (B), switch (S) and attention-switch (A+S) architectures, in the instances of chord change.

In an attempt to further study the behavior of the architectures, we repeat the above experiment, for the baseline network (B), the one using a chord switch detector (S) and the one further utilizing attention (A+S), setting the input chord progression length equal to 32 timesteps. We do not only report on the percentage of instances where the chord prediction is absolutely correct (Acc.), but we further calculate the percentage of instances where the chord prediction is correct with respect to either the chord chroma (C. Acc.), or the chord type (T. Acc.). Again, we differentiate between overall accuracy percentages, and accuracy percentages in chord shift instances, since the second case is significantly more challenging, while taking into account both top-1 and top-5 accuracy percentages.

The results of the above analysis are presented in Tables 1, 2. We may conclude that it is generally easier for the tested network architectures to infer the type of the following chord, compared to its chroma. This is especially apparent for the architectures not utilizing an attention mechanism, since their prediction accuracy regarding the chord type is comparable - and, in the case of the baseline, superior - to the one that utilizes it. This surprising result could be explained, given that the baseline architecture has been trained using the full dataset, where the chord type remains the same in the majority of instances, which is actually the prevalent behavior in the chord shift instances as well. Otherwise, there is no significant trend difference between the top-1 and top-5 accuracy percentages. Finally, we observe that the comparative advantage of the attention-utilizing network is increased when taking into account only the instances where the chord progression shifts into a different chord.

5. SUBJECTIVE EVALUATION

In order to further study the long-term behavior of our system, we designed a simple experiment. We deployed three different architectures - the baseline (B), the baseline utilizing a switch mechanism (S), and the attention-utilizing architecture (A+S) to our interactive environment, using the tensorflow.js[1] framework. Three distinct input chord sequences were also provided as seeds for initializing the networks. A total of 12 users, 9 male and 3 female, of an average age of 27.33 years, out of whom 6 had prior musical knowledge, tested our system for 10 minutes each, using all possible network configurations, as well as a variety of potential seed sequences for each configuration. The network architecture was changed and re-initialized whenever either the users felt satisfied with their interaction with the system, or the architecture converged to a specific set of proposed chords. Afterwards, they were asked to evaluate the output of the system with regards to the perceived musical coherence of the provided song, as well as the variety of the suggested chords over time, in 5-point Likert scales.

Architecture	Musical Coh.	Variety
B	**3.58**	1.83
S	3.33	3.08
A+S	3.08	**3.67**

Table 3. Results of the subjective evaluation of our system with regards to perceived musical coherence and variety of proposed chords, using a 5-point Likert scale.

The results are presented in Table 3. In the case of the subjective evaluation, the use of a neural attention mechanism in the chord generator largely increased the variety of chords offered, steering clear of proposing the same chord progressions in the long term. This issue was especially evident in the case of the simple LSTM architecture, whereas it was comparatively avoided in the architecture utilizing the switch mechanism - likely because of the different training subset used for the generator network in this case. We further note that the two architectures not utilizing the attention mechanism yielded, albeit not significantly, more coherent chord progressions. Finally, it was observed that the general behavior of the architectures was relatively independent of the initial seed choices.

6. CONCLUSIONS AND FUTURE WORK

In this work, we studied the use of LSTM-based architectures utilizing attention modules in the tasks of chord prediction and chord progression generation. Whilst the results are relatively satisfying with regards to the short-term prediction ability of our system, there is still ground to cover in long-term chord progression generation, as evidenced by the subjective evaluation results. Future expansions of the current work could involve incorporation of recent breakthroughs in natural language modeling techniques into chord prediction. Furthermore, the final mapping between the generator's output and the proposed chords could be done by using a different chord representation or distance metric, so that the chord proposals are more perceptually consistent. Finally, the whole generative system could be more extensively evaluated with regards to adhering to musical patterns.

7. ACKNOWLEDGEMENTS

The authors would like to thank the participants in the subjective evaluation of our system for their time and feedback, Nikos Gkanatsios and Kosmas Kritsis for the fruitful discussions on the content of this paper, as well as the reviewers for their useful comments.
This work is supported by the *iMuSciCA* project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 731861.

[1] www.tensorflow.org/js

8. REFERENCES

[1] M. Allan and C.K.I. Williams, "Harmonising chorales by probabilistic inference.," in *Proc. (NIPS-05)*, Vancouver, BC, Canada, 2005.

[2] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription," in *Proc. Int'l Conf. on Machine Learning*, Edinburgh, UK, 2012.

[3] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, "Music transcription modelling and composition using deep learning," in *Proc. Conf. on Computer Simulation of Musical Creativity*, Huddersfield, UK, 2016.

[4] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. AAAI Conf. on Artificial Intelligence*, New Orleans, LA, USA, 2018.

[5] C. Donahue, I. Simon, and S. Dieleman, "Piano genie," in *Proc. Int'l Conf. on Intelligent User Interfaces (IUI'19)*, Los Angeles, CA, USA, 2019.

[6] G. Hadjeres, F. Pachet, and F. Nielsen, "Deepbach: a steerable model for bach chorales generation," in *Proc. Int'l Conf. on Machine Learning*, Sydney, Australia, 2017.

[7] J. A. Hennig, A. Umakantha, and R. C. Williamson, "A classifying variational autoencoder with application to polyphonic music generation," *arXiv preprint arXiv:1711.07050*, 2017.

[8] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proc. Int'l Conf. on Machine Learning*, Stockholm, Sweden, 2018.

[9] W. Brent, "The gesturally extended piano.," in *Proc. Conf. on New Interfaces for Musical Expression (NIME-12)*, Ann Arbor, MI, USA, 2012.

[10] A. Rosa-Pujazon, I. Barbancho-Perez, L. Tardon, and A. Barbancho-Perez, "Conducting a virtual ensemble with a kinect device," in *Proc. Sound and Music Computing Conference*, Stockholm, Sweden, 2013.

[11] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[12] C. Garoufis, A. Zlatintsi, K. Kritsis, P.P. Filntisis, V. Katsouros, and P. Maragos, "An environment for gestural interaction with 3d virtual musical instruments as an educational tool," in *Proc. 27th European Signal Processing Conf.(EUSIPCO-19)*, A Coruna, Spain, 2019.

[13] E. J. Humphrey, D. Turnbull, and T. Collins, "A brief review of creative mir," *ISMIR Late-Breaking News and Demos*, 2013.

[14] M. Unehara and T. Onisawa, "Music composition by interaction between human and computer," *New Generation Computing*, vol. 23, no. 2, pp. 181–191, 2005.

[15] M. Kaliakatsos-Papakostas, A. Gkiokas, and V. Katsouros, "Interactive control of explicit musical features in generative lstm-based systems," in *Proc. Audio Mostly (AM'18) on Sound in Immersion and Emotion*, Wrexham, UK, 2018.

[16] R. Fiebrink and H. Scurto, "Grab-and-play mapping: Creative machine learning approaches for musical inclusion and exploration," in *Proc. of the Int'l Computer Music Conference.*, Utrecht, Netherlands, 2016.

[17] J. Shao, J. McDermott, M. O'Neill, and A. Brabazon, "Jive: A generative, interactive, virtual, evolutionary music system," in *European Conf. on the Applications of Evolutionary Computation*, Istanbul, Turkey, 2010.

[18] Elliot Waite, "Generating long-term structure in songs and stories," in *Magenta Blog*, 2016.

[19] H.-W. Dong and Y.-H. Yang, "Convolutional generative adversarial networks with binary neurons for polyphonic music generation," in *Proc. Int'l Society for Music Information Retrieval Conference (ISMIR-18)*, Paris, France, 2018.

[20] D. R. Tuohy and W. D. Potter, "A genetic algorithm for the automatic generation of playable guitar tablature.," in *Proc. Int'l Computer Music Conference (ICMC-05)*, Barcelona, Spain, 2005.

[21] M. McVicar, S. Fukayama, and M. Goto, "Autorhythmguitar: Computer-aided composition for rhythm guitar in the tab space," in *Proc. Int'l Computer Music Conference (ICMC-14)*, Athens, Greece, 2014.

[22] J. A. Burgoyne, J. Wild, and I. Fujinaga, "An expert ground truth set for audio chord recognition and music analysis.," in *Proc. Int'l Society for Music Information Retrieval Conference (ISMIR-11)*, Miami, FL, USA, 2011.

[23] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. Int'l Conf. on Learning Representations (ICLR-15)*, San Diego, CA, 2015.