Calliope: A Co-creative Interface for Multi-Track Music Generation

Renaud Bougueng Tchemeube
rbouguen@sfu.ca

Jeff Ens
jeff_ens@sfu.ca

Philippe Pasquier
pasquier@sfu.ca

School of Interactive Arts and Technology
Simon Fraser University
Surrey, BC V3T 0A3, Canada

Abstract

With the rise of artificial intelligence in recent years, there has been a rapid increase in its application towards creative domains, including music. There exist many systems built that apply machine learning approaches to the problem of computer-assisted music composition (CAC). Calliope is a web application that assists users in performing a variety of multi-track composition tasks in the symbolic domain. The user can upload (Musical Instrument Digital Interface) MIDI files, visualize and edit MIDI tracks, and generate partial (via bar in-filling) or complete multi-track content using the Multi-Track Music Machine (MMM). Generation of new MIDI excerpts can be done in batch and can be combined with active playback listening for an enhanced assisted-composition workflow. The user can export generated MIDI materials or directly stream MIDI playback from the system to their favorite Digital Audio Workstation (DAW). We present a demonstration of the system, its features, generative parameters and describe the co-creative workflows that it affords.

Introduction

The development of computer-assisted composition (CAC) systems is a research activity that dates back to at least the works by IRCAM on OpenMusic (Assayag et al. 1999). CAC is a field that is concerned with developing systems that are capable of automating partially or completely the process of music composition. There exist several compositional tasks a system can address: multi-track pattern generation, multi-track complete generation, rhythm generation, harmonization, chord progression generation, melody generation, interpolation, form-filling, orchestration and interpretation. Many machine learning-based (ML) systems have been developed for computer-assisted composition including: Flow Machines (Pachet 2004), Style Machine (Anderson, Eigenfeldt, and Pasquier 2013), Magenta Studio (Roberts et al. 2019), Manuscore (Maxwell et al. 2012), Morpheus (Herremans and Chew 2017); demo systems such as Sornting, DrumVAE (Thio et al. 2019), DeepDrum (Makris, Kaliakatsos-Papakostas, and Kermanidis 2018) and commercial systems such as AIVA [1], Spliqs [2] and Melody Sauce [3]. Magenta Studio, DrumVAE and Sornting deploy algorithms based on the MusicVAE model (Roberts et al. 2018). DeepDrum proposes an adaptive neural network model for better capturing drum rhythms. Flow Machines and Style Machine employ Markov models. Manuscore uses a cognitive architecture and Morpheus combines a tensor model with constraint rules. Finally, AIVA, Spliqs and Melody Sauce employ proprietary algorithms; the first two for generating conventional multi-track music, and the last one, for melody creation. Calliope differentiates itself by using a Transformer model called the Multi-Track Music Machine (MMM). MMM, trained on half a million MIDI files (Ens and Pasquier 2020a), offers genre-agnostic batch-enabled generative capabilities. Its rich multi-level attribute controls combined with bar infilling enables to tackle many composition tasks at once. Calliope has been released publicly and is being used by a variety of composers for artistic purposes and in the context of usability and acceptability evaluation studies. The project is available at **https://metacreation.net/calliope**.

System Description

Building on top of Apollo, our interactive web environment that makes corpus-based music algorithms usable for training and generation via a convenient graphical interface (Tchemeube, Ens, and Pasquier 2019), Calliope (Figure 1) is narrowed down for MIDI manipulation in the browser, generative controllability of the MMM model, batch generation of partial or complete multi-track compositions and interoperability with other MIDI-based systems. The aim is to enable users to effectively co-create with a generative system. Calliope is built in Node.js, the Web stack (HTML, CSS, Javascript) and MongoDB. It is made interoperable with the MMM pre-trained model via a Python process runtime.

MIDI Viewing and Playback

MIDI notes from any uploaded MIDI file can be visualized in a piano roll format (Figure 2). Metadata info such as the MIDI channel number and assigned MIDI instrument can also be viewed and edited. The MIDI player supports the General MIDI (GM) standard for MIDI playback and the capacity to select from a list of soundfonts.

[1] https://www.aiva.ai/
[2] https://www.spliqs.com/
[3] https://www.evabeat.com/

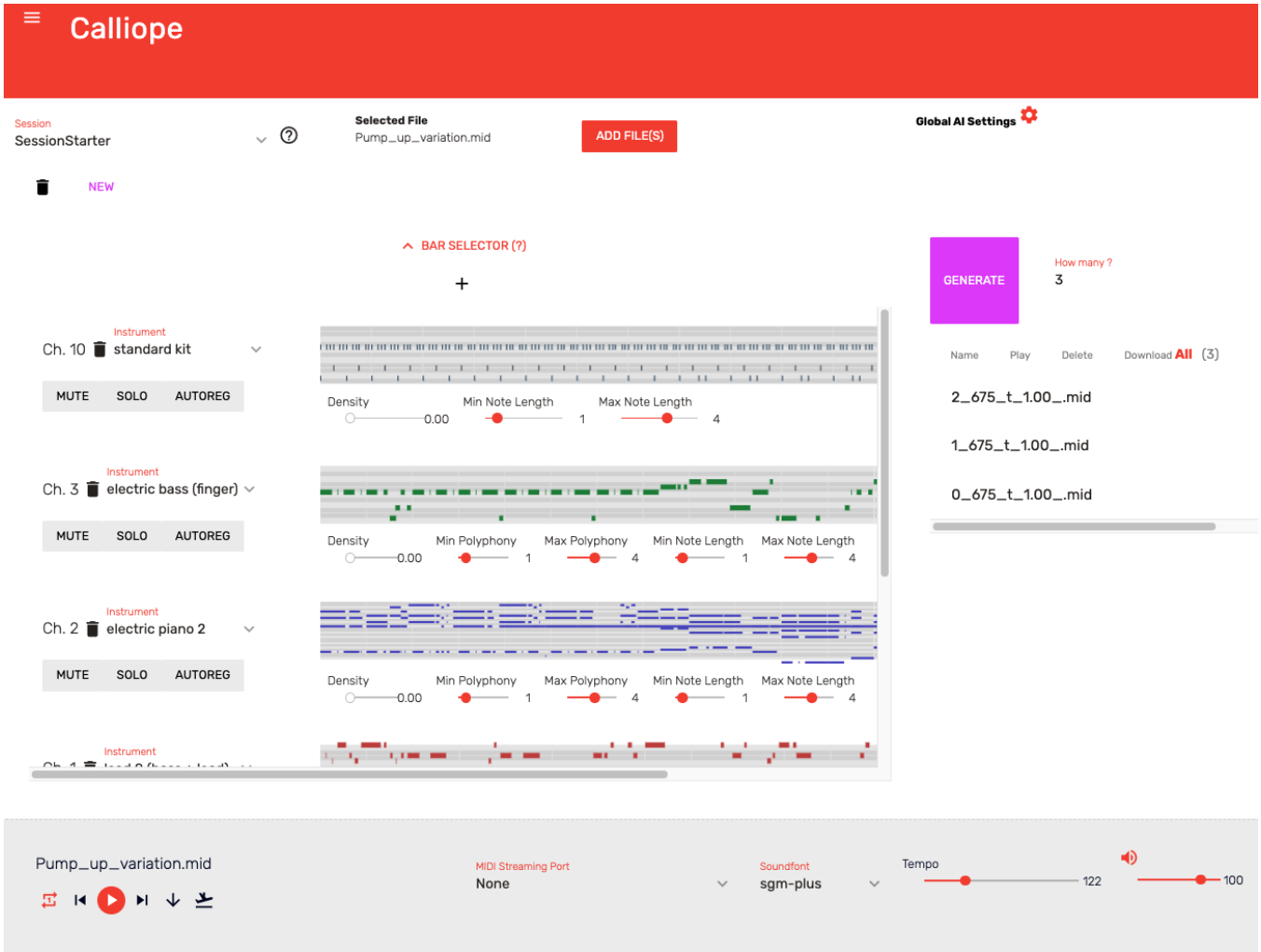

Figure 1: Calliope's Interface

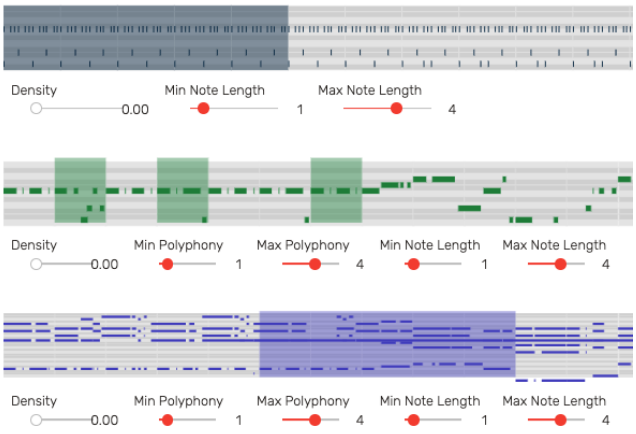

Figure 2: Multi-Track Piano Roll with Bar In-Filling

Conditioned Music Generation

Generation is achieved using the Multi-Track Music Machine (Ens and Pasquier 2020a). Because of its design which uses bar selection, and a set of *global* and *local* attribute controls, the model can accommodate a variety of compositional tasks.

Bar Selection MMM's primary mode of generation is *bar in-filling*. The model can generate note patterns for bars in a given multi-track MIDI file. A subset of bars across the multi-track content can be selected for generation by visually highlighting them (Figure 2). It is also possible to temporarily edit the MIDI file by deleting or adding tracks. This is useful to perform generation on a subset of the MIDI tracks or to generate a new track for a given MIDI file. Generated music for a particular subset of bars is constrained on musical information that precedes those bars (within the given track) and on musical information found within the neighboring tracks.

Global Parameters MMM offers the following *global* (model-level) generation parameters (Figure 3):

- **Temperature** [0.8, 1.2]: Also called typicality, this *float* value determines how much the structure of the generated MIDI content is closer (conservative) or farther (experimental) to what the MMM model is most likely to generate. Technically, it corresponds to the the temperature in the sampling of the neural network.
- **Polyphony Hard Limit** {1-6}: The global maximum number of simultaneous notes the system can generate at any given moment.
- **Percentage** {0-100}: This parameter controls how much of the existing MIDI content is preserved or replaced by the generation. This is done based on the number of tracks per step and bars per step. For example, for tracks per step and bars per step each 4, and percentage at 25, the model will process only 4 out of 16 bars to be generated at each generation step.
- **Model Dimensions** {1-8}: The dimension of the model in bars. This is the window size used by the model to process MIDI input data for generation. The default value is 4 corresponding to a 4-bar window.

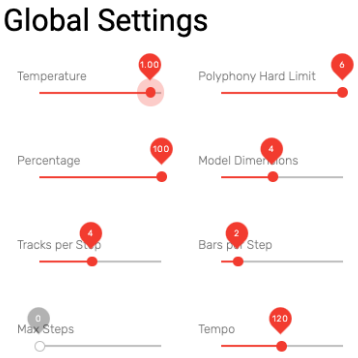

Figure 3: MMM's Global Parameters

- **Tracks per Step** {1-8}: Number of tracks being processed at each generation step. The default value is 4.
- **Bars per Step** {1-8}: The number of bars processed within each track at each generative step. The default value is 2.
- **Max Steps** {0-8}: The maximum number of generation steps. This value can be used to avoid memory overload. When it is set to zero, it is ignored by the system.
- **Tempo**: The resulting tempo for the generated output as a positive *integer* value.

Track Parameters In addition to model-level parameters, MMM offers a set of *local* (track-specific) music-based generation parameters (Figure 2). Such parameters are available to be specified for each track of a given MIDI file. They are:

- **Instrument Type**: The type selector is composed of a set of 128 instrument types and 8 instrument groups following the MIDI GM Standard [4]. It conditions MMM to generate in the style of the chosen instrument. For example, if *violin* is selected, MMM generates a MIDI pattern to be played by a violin instrument. This is especially convenient to differentiate the *percussion* group (e.g. drums) vs other instrument track types (e.g. *guitar*, *strings*, *synth lead* groups).
- **Note Density** [0-10]: The number of notes generated per bar size. The higher this value, the more likely the model is to generate bars with a high total number of notes. A value of zero means that the note density is set at random by the model for each generation request.
- **Polyphony Range** {0, 1, 2, 3, 4, 5, 6}: the number range of simultaneous notes used by the model as a soft constraint for generation. The upper limit of this parameter is automatically overriden by the value of the "Polyphony Hard Limit" global parameter.
- **Note Duration Range** {Any, 1/32, 1/16, 1/8, 1/4, 1/2, Whole}: Note duration values are defined in accordance

[4] https://en.wikipedia.org/wiki/General_MIDI

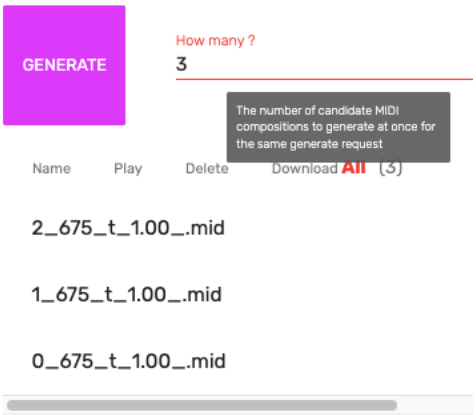

Figure 4: Batch Number for Generation

to the Western music notation. For example, 1/16 corresponds to a note duration equivalent to a sixteenth note.

Batch Generation of Music Outputs

Batch generation of musical outputs is implemented by passing a *batch_size* parameter (Figure 4) to the MMM Python interface which offers batch support natively. The ability to batch generate means that the user can quickly explore alternatives, including generating from a previously generated output, for a given set of control parameters. We have tested generation of 5, 10, 100, 500, 1000 music samples at a time. These generations can be done within 3 seconds to 10 minutes on an average computer depending on the total note density of the music input.

Ranking

It is possible to rank a collection of generated MIDI files against a selected one. This is useful to informally evaluate the quality of the model generations. We employ a ranking algorithm which statistically quantifies the similarity of a generated output MIDI file against the set of other MIDI files (Ens and Pasquier 2020b). This enables assessing accuracy or reliability of the MMM model for style imitation tasks. From an interaction point-of-view, it helps the user explore the variability in similarity among MIDI files and effectively apply filter operations on the set of files. This is especially useful in the context of large set of generated files (e.g. set of 50 files and up).

Co-Creative Interaction

In terms of co-creation, the user can configure multiple attribute controls for generation (instrument type, node density, polyphony range, note length range, bar selection within a piece). Those controls set the creative context for the system to generate, allowing the user to steer the generative behavior of the model and guide the composition process. The system generates new musical phrases by outputting multi-track polyphonic sequences of notes for the set of selected bars and in accordance to the attribute control values. The user listens and analyzes the resulting output

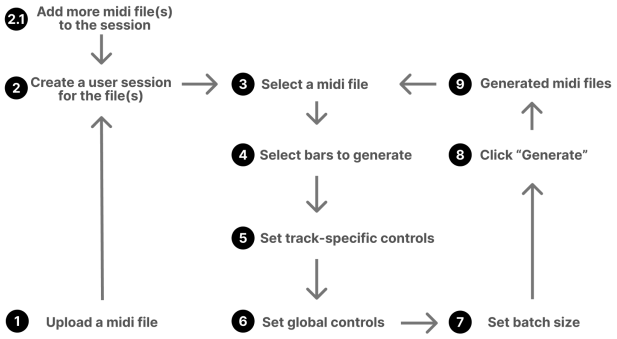

Figure 5: Compositional Workflow in Calliope

and updates the generation request accordingly. The steps involved in Calliope's interactive workflow are shown in Figure 5. Generation happens within an interactive context defined by a *user session* (step 2). The user session itself is defined by a seed MIDI file, which is used to kick-off the first generation. The connection from steps 9 to 3 highlights how generated outputs can themselves later be fed back into the system as seed MIDI files for new user sessions. This enables more complex workflows for the user within Calliope.

MIDI Streaming

Additionally, it is possible for the user to stream MIDI playback to their favorite DAWs to assign playback to their own project session instrumentation. Calliope can be integrated with the user's digital studio (e.g. Ableton) via a MIDI port accessible in the MIDI player. This provides a unique opportunity for the user to interface their native environment with a generative system. Users can stream playback of generated MIDI files to their preferred instrumentation and sounds, including applying their existing preferred signal chains to the live output audio stream. This opens up new areas for workflow experimentation given a computer-assisted composition framework. Alternatively, they can download the MIDI files from Calliope and import them back into new or existing DAW project sessions.

Conclusion

We presented the Calliope system, a co-creative interface for multi-track music generation. We presented its features including the ability to view and play MIDI files, the ability to select bars to guide partial generation, and complete set of global (model-level) and local (track-specific) controls and how their combination allows users to tackle a broad range of compositional tasks. We situated our system with respect to other existing CAC systems and discussed the co-creative aspect of the system along with the compositional workflow it affords. The Calliope system is at the beta phase and we are working on its next version. More future work includes an ongoing evaluation study of the system along human factors including usability, user experience on feeling of trust, authorship, controllability and measured of technology acceptance among amateurs and professional composers.

Author Contributions

Bougueng R. T. was in charge of writing the manuscript and developed a significant part of the system. Ens J. developed the integration of the algorithm used by the system and assisted in making its use functional. Pasquier P. supervised the entire research process and provided direction and guidance to the project implementation. All authors participated in the writing of this manuscript and are listed in alphabetical order.

Acknowledgments

We would like to acknowledge SSHRC, NSERC and MITACS for their contribution in helping fund this research. We would also like to acknowledge Tara Jadidi as a code contributor to Calliope MIDI Viewer/Editor and bar selection feature.

References

Anderson, C.; Eigenfeldt, A.; and Pasquier, P. 2013. The generative electronic dance music algorithmic system (gedmas). In *Proceedings of the Second International Workshop on Musical Metacreation (MUME 2013) 2013*.

Assayag, G.; Rueda, C.; Laurson, M.; Agon, C.; and Delerue, O. 1999. Computer-assisted composition at ircam: From patchwork to openmusic. *Computer Music Journal* 23(3):59–72.

Ens, J., and Pasquier, P. 2020a. Mmm: Exploring conditional multi-track music generation with the transformer. *arXiv preprint arXiv:2008.06048*.

Ens, J., and Pasquier, P. 2020b. Quantifying musical style: Ranking symbolic music based on similarity to a style. *arXiv preprint arXiv:2003.06226*.

Herremans, D., and Chew, E. 2017. Morpheus: generating structured music with constrained patterns and tension. *IEEE Transactions on Affective Computing* 10(4):510–523.

Makris, D.; Kaliakatsos-Papakostas, M.; and Kermanidis, K. L. 2018. Deepdrum: An adaptive conditional neural network. *arXiv preprint arXiv:1809.06127*.

Maxwell, J. B.; Eigenfeldt, A.; Pasquier, P.; et al. 2012. Manuscore: Music notation-based computer assisted composition. In *Proceedings of the International Computer Music Conference (ICMC 2012)*.

Pachet, F. 2004. On the design of a musical flow machine. *A Learning Zone of One's Own,* pp. 111–134.

Roberts, A.; Engel, J.; Raffel, C.; Hawthorne, C.; and Eck, D. 2018. A hierarchical latent vector model for learning long-term structure in music. *arXiv preprint arXiv:1803.05428*.

Roberts, A.; Engel, J.; Mann, Y.; Gillick, J.; Kayacik, C.; Nørly, S.; Dinculescu, M.; Radebaugh, C.; Hawthorne, C.; and Eck, D. 2019. Magenta studio: Augmenting creativity with deep learning in ableton live. *In Proceedings of 7th International Workshop on Musical Metacreation (MUME 2019)*.

Tchemeube, R. B.; Ens, J.; and Pasquier, P. 2019. Apollo: An interactive environment for generating symbolic musical phrases using corpus-based style imitation. *In Proceedings of 7th International Workshop on Musical Metacreation (MUME 2019)*.

Thio, V.; Liu, H.-M.; Yeh, Y.-C.; and Yang, Y.-H. 2019. A minimal template for interactive web-based demonstrations of musical machine learning. *arXiv preprint arXiv:1902.03722*.