DuetHLSTM: A Hierarchical LSTM Model for Two-Track Music Generation

Jocelyn Huang
jocelynh@andrew.cmu.edu

Menglan Ji
menglanj@andrew.cmu.edu

Daniel Martin
dlmartin@andrew.cmu.edu

Abstract

Music generation presents challenges due to the necessity of harmonic, rhythmic, and temporal structure in realistic music. Many neural network models for music generation use recurrent networks to address recurring elements in music. We present a hierarchical LSTM model to further address inter- and intra-track dependencies in order to generate realistic polyphonic multi-track music, with separate but harmonious melodies generated in each instrumental track. We train our model on the multi-track piano rolls in the Lakh Pianoroll Dataset, and we conclude that our model produces relatively realistic and coherent musical sequences.

1 Introduction

The generation of realistic and pleasing music is an interesting and challenging problem due to its subjectivity and nuances in what is considered "good" music. Though the evaluation of music is highly subjective, a plausible piece of generated music should generally have clear harmonic and rhythmic structure, regularity over time, and, for multi-track pieces, coherence between tracks.

First, the model should take into consideration relative differences between dimensions such as pitch and timing in order to be able to generate harmonic and rhythmic music. Second, musical pieces have recurring elements on various levels. Thus, a music generation model should be aware of previous elements, being able to repeat and vary them in a sensible way. Generating multiple tracks of music introduces the additional challenge of temporal dependencies between the two tracks as well as tonal dependencies, since two separate melodies being played independently often sounds inharmonious. In addition, realistic music also integrates polyphony, where multiple notes may be played at once in a single track, such as in a chord, so there are also intra-track melodic dependencies.

We propose DuetHLSTM, a sequence generation network model architecture that makes use of a hierarchical LSTM to generate polyphonic two-track music, with one track being a melody and the other being an accompaniment. We address the issues of time dependencies with the recurrent advantages of an LSTM, and we address the track dependencies between melody and harmony with the hierarchical structure. We train on 102,378 unique multi-track, binary-valued piano rolls from the Lakh Pianoroll Dataset[1]. Experimental results show that the binary-valued piano rolls we generate are generally realistic and structurally similar to the training data, and comparably favorably against previous work trained on the same dataset.

[1]https://salu133445.github.io/lakh-pianoroll-dataset/

2 Background

We previously worked on adapting the MuseGAN network to use a hierarchical LSTM structure. MuseGAN [2] is a convolutional GAN model that generates a fixed number of bars of multi-track music in piano roll format, either from scratch (inference) or as accompaniment given an existing track (interpolation). The authors have also explored whether adding binary (instead of real-valued) output neurons would aid in smoothing the output in an updated model called BMuseGAN [3].

The MuseGAN model was trained on the Lakh Pianoroll Dataset. It uses a shared generator to create an overall representation that is passed to multiple private generators that then generate each separate track of music. It also includes a refiner, which converts the real-valued outputs of the generator to binary values that determine whether each note is kept or not in the final pianoroll. The discriminator in MuseGAN consisted of several private convolutional networks, as well as a shared CNN, mirroring the structure of the generator. In addition to the raw notes, the discriminator also uses metrics calculated on the music like the chroma stream and onset/offset stream.

MuseGAN was related to an earlier model called MidiNet [5], which also used CNNs arranged as a GAN. The input music was divided into bars, which were used as data points. MidiNet used a conditional mechanism to generate either from scratch or conditioned on prior bars.

For our midterm report, we ran inference on a pretrained MuseGAN model available from the authors on GitHub[2], and picked several of the generated bars of music to convert to MIDI for rating. We then asked five human subjects to rate 10 bars of the music on a 1-5 scale, where 1 is poor and 5 is excellent, taking into mind qualities such as rhythm, structure, harmony, and enjoyment of the music. The average rating across all clips was 2.9. While the human subjects observed that the generated music had some rhythm, they also noticed that parts of the generated music sounded like hammering away at a piano and generally did not have much melodic structure.

3 Related Work

In addition to MuseGAN, many other models for music generation have been proposed, with different methods of dealing with temporal and tonal structure. In this section we briefly summarize other methods we drew inspiration from for our model.

Due to the naturally sequential aspect of music, Huang et al. [4] introduce an algorithm for the Transformer, a sequence model based on self-attention, with a relative attention mechanism. Their modifications allow them to produce regular phrasing in their generated music and reduce memory requirements. They represent the music using a language-modeling approach, representing each piece as a sequence of discrete token drawn from a vocabulary. The Transformer model relies on positional sinusoids to represent timing, but Huang et al. learn a separate relative position embedding using a memory-efficient "skewing" algorithm.

They ran experiments on the J.S. Bach Chorales and MAESTRO datasets. They were able to show improved note-wise losses on the validation datasets. In addition, human listeners were also selected their model's generated music as more musical, in comparison to previous models.

Yu and Varshney [7] provided a different approach to this problem. They also recognized the importance of hierarchical structure in the learning process. Their model used a collaborative approach: the generator (student) started out generating at random, and at each iteration, the discriminator (teacher) had the student correct the feature that had the greatest difference from the input style. From this collaborative style, it was possible to recover concrete "rules" for generating data. This approach was more open-ended than a GAN, so the authors introduced information-theoretic metrics for guiding training, and compared the extracted rules to commonly-accepted rules of music theory.

Chu, Urtasun, and Fidler [1] proposed a hierarchical recurrent sequence generation music generation model for generating a non-polyphonic melody and two backing tracks, one for chords (polyphonic) and one for drums (non-polyphonic). They trained their model on pop music, but incorporated additional features such as a target scale that the generated music should be based on, a human-generated melody profile that represented the music flow that should be followed, and several

[2]https://github.com/salu133445/musegan

lookback features relating to previous notes that overall give the model more structure to work with when generating each set of notes.

At each time step, their model picks a single note between C3 to C6 (one of 36 possible notes) or silence for the melody, using an LSTM with skip-connections conditioned on the scale. The input to this layer is a one-hot encoding of the previous note as well as the melody profile and lookback features. They denote this as the key layer, since its output determines what notes are played in the main melody, and is what each subsequent layer is based upon. Next, the duration layer (which determines the duration of the melody notes) as well as the two backing track layers are conditioned on the key layer in order to form the raw output of the model. They then perform what they call temporal alignment on the output of the model, which regularizes the changes in notes to match up with the start of the measures in the music.

They claim that their model performed well in evaluations across 27 human subjects comparing their generated output against music generated by Google Magenta, though they mentioned that the human subjects remarked that the accompanying tracks do not match up perfectly with the main melody even after temporal alignment.

4 Methods and Model

4.1 Dataset

We used the Lakh Pianoroll dataset for training, which is the same dataset as was used in MuseGAN. Each sample from this dataset consists of four bars of music, 96 time steps per bar, and 84 possible notes per track across five tracks, yielding a $4 \times 96 \times 84 \times 5$ size tensor for each sample.

However, since we generate only two tracks of music rather than five, we reformatted the dataset to only use the piano and strings tracks, which we treated as the melody and accompaniment, respectively. We also subsampled to use eight time steps per bar since a vast majority of the samples had a smallest subdivision of notes at least one-eighth of a bar, and we saw better results with subsampled training data. Without subsampling, predicting the same note as the input would be correct a vast majority of the time, leading to unchanging notes over four bars. We also combine the bars and timesteps per bar dimensions, which leads to a music sample shape of $32 \times 84 \times 2$ (total timesteps, then number of possible notes per track, then number of tracks).

See Figure 6(a) in the Results section for three examples of four-bar training data pianorolls.

4.2 Previous Approaches

Our originally proposed model was a generative adversarial network that utilized a hierarchical LSTM generator and a convolutional discriminator. We had hoped to create a generator that would learn temporal and track dependencies and a discriminator that learned to find an overall structure; however, we quickly ran into a few issues in structuring and training the GAN.

Our first problem came from the need to binarize the output in order to create pianorolls. Each pianoroll is a sequence, where each timestep is represented by a vector of 84 pitch values. The ith entry is set to 0 if the pitch is not played at that timestep, or 1 if it is. As such, all the training data was binary-valued, and any generated real values would need to be binarized to create a pianoroll that made sense. However, the rounding operation is not differentiable, and though we then tried the original MuseGAN approach of just passing in the real values to the discriminator, and rounding when generating pianorolls, we were not able to achieve success.

Another problem came from the need to call the generator multiple times and repeatedly append outputs in order to create a full pianoroll with 4×8 timesteps before passing the output to the discriminator due to the sequential structure of our music generation model. This structure of GAN is generally not supported by default by any framework that we could find in a reasonable amount of time, and the custom model that we put together in Keras did not manage to train at all. As such, we decided to modify our generator architecture to train in a sequence prediction task.

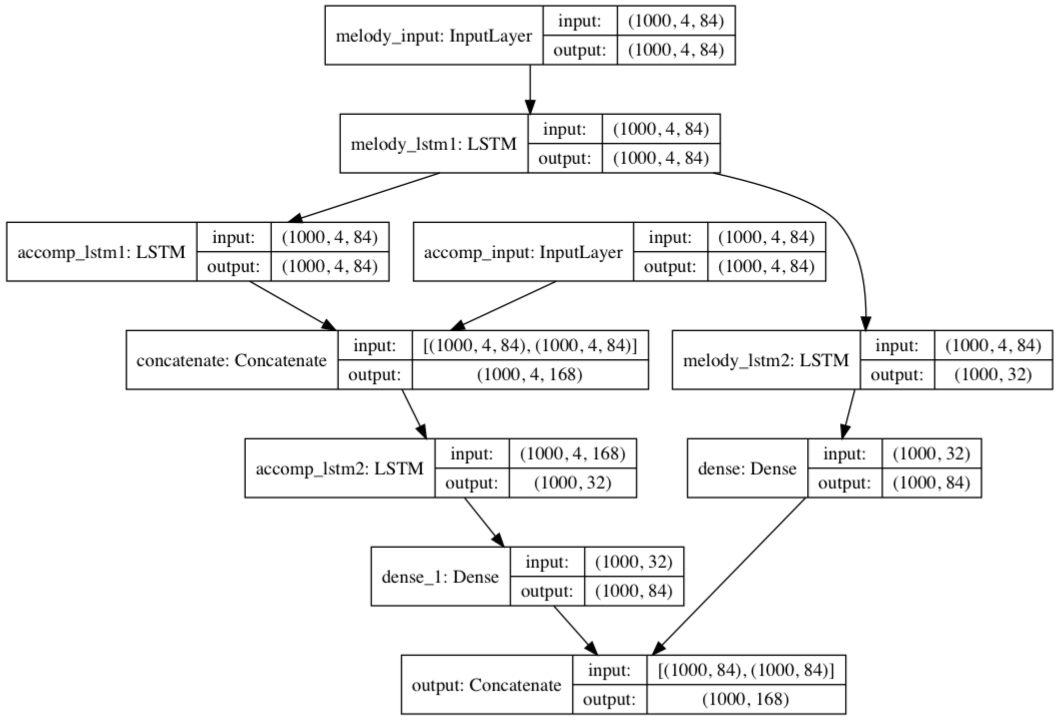


Figure 1: Visualization of DuetHLSTM architecture with batch size 1000.

4.3 DuetHLSTM

DuetHLSTM, our music generation model, is a sequence generation network that was inspired by the hierarchical structure from Chu, Urtasun, and Fidler [1]. Our goal was to leverage the sequence-sensitive advantages of an LSTM to address temporal relationships in music generation, while utilizing the advantage of a stacked LSTM structure to tie together tonal and temporal dependencies of the accompaniment track to the melody track.

However, unlike the model created by Chu, Urtasun, and Fidler, DuetHLSTM does not use any other features besides the previous four timesteps of music, and we do not perform any additional processing or temporal alignment of the generated music besides conversion to MIDI for evaluation. We also use a different training set and therefore our output is formatted differently, with the goal of creating polyphonic music over both tracks instead of just one. Lastly, we pass in the full previous timesteps of pianorolls, with both the melody and the accompaniment, whereas Chu, Urtasun, and Fidler's model takes only the melody notes. We do this because we found that not including at least some accompaniment information led to the model often ignoring the accompaniment track entirely, predicting zero activation for any accompaninent output.

DuetHLSTM takes as input the four previous timesteps of music (half of a bar), and tries to predict the next timestep of polyphonic music in pianoroll form, choosing "on" or "off" values of 84 possible notes over both melody and accompaniment tracks. In order to generate multiple timesteps of music, the model is run multiple times, passing in the previous output as an input, and the final output consists of all timesteps' outputs concatenated along the first, temporal dimension.

First, the model passes in the previous four timesteps of a melody pianoroll through an LSTM layer, and the resulting hidden states are then passed to both a secondary melody LSTM and an accompaniment LSTM. A secondary accompaniment LSTM takes as input the hidden states of the first accompaniment LSTM, combined with the previous four timesteps of the accompaniment pianoroll. Both outputs of the secondary melody and accompaniment LSTMs are passed through fully connected layers with softmax activation, and the final outputs are concatenated into one vector of size 168 (two times the 84 possible output notes for each track).

Table 1: Pitch and rhythm features of music from Lakh, MuseGAN, and DuetHLSTM.

	PC	PR	NC
Lakh	56	65	12
MuseGAN	16	29	7
DuetHLSTM	76	80	12

Figure 1 shows a visualization of the model architecture, along with each layer's corresponding input and output dimensions for a batch size of 1,000.

Originally, we had tried using mean-squared error loss, treating music generation as a multi-dimensional regression problem with a range between zero and one, but we found that it did not quite fit the task of binary prediction for each of 168 notes. The best results came from training the model using binary cross-entropy loss, in a sense treating whether each note was "on" or "off" as a multi-label binary classification task. Though it does treat each output note as a separate "label" from the other notes, it seems that the model still manages to learn correlations between notes and that sound harmonious when played together.

The full code for DuetHLSTM, along with sample generated MIDI files at 1, 10, 20, 30, and 40 epochs of training, are available on GitHub[3].

5 Results

The automatic evaluation of generated music is a challenge, as the inherently subjective nature of creative mediums like music makes it difficult to evaluate the output of generative modeling. Human judges provide the best evaluation of music, but there is still a measure of subjectivity.

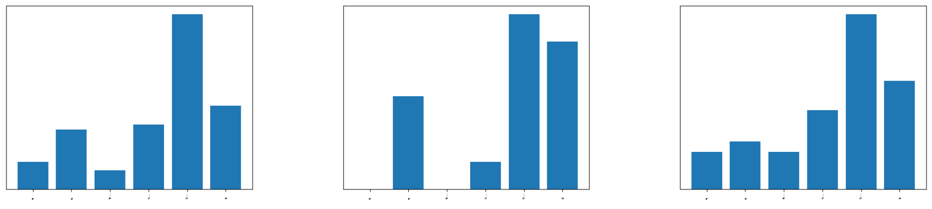
Yang and Lerch [6] propose several objective metrics that can be used to evaluate features of generated music and to therefore analyze and compare generated music in a more systematic way. These metrics include both rhythm-based and pitch-based metrics:

- Note count (NC): The number of different note lengths in a sample.
- Note length histogram (NLH): Representation of the occurrence of the notes in a sample. Here we only allow the defined notes [whole, half, quarter, 8th, dot half, dot quarter].
- Note class transition matrix (NCTM): A histogram-like representation of the transition between each ordered pair of notes in the sample, again only using the defined notes.
- Pitch count (PC): The number of different pitches in a sample.
- Pitch class histogram (PCH): An octave-independent chromatic representation of pitches in a sample.
- Pitch class transition matrix (PCTM): A histogram-like representation of the transition between each ordered pair of notes in the sample.
- Pitch range (PR): The difference between the highest and lowest pitch in the sample.

Table 1 shows the pitch count, pitch range, and note range of the Lakh dataset, which is the training set, MuseGAN, and DuetHLSTM, taken over a sample of 200 bars of music. We see that the Lakh data and DuetHLSTM both have 12 distinct note lengths, while MuseGAN only has 7. In addition, the pitch count and pitch ranges are both relatively high for the Lakh data and DuetHLSTM-generated music, but the MuseGAN-generated music has less than a third of the generated pitches that the other two have, and only half the pitch range. Thus, it appears that DuetHLSTM was able to better learn the features of the original training data.

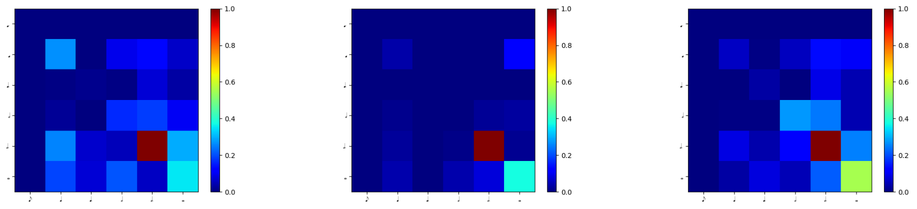
This is also supported in the histogram visualizations. The NLH and NCTM visualizations in Figures 2 and 3 show that the MuseGAN-generated music was not entirely dissimilar to the Lakh training data. All three NCTMs show that the dot half note has the most number of appearances in the music, but the NLH and NCTM of DuetHLSTM has a more similar distribution to that of the training data.

[3] https://github.com/redoctopus/Music-Generation



(a) Lakh dataset. (b) MuseGAN. (c) DuetHLSTM.

Figure 2: Visualization of NLH from three sources.



(a) Lakh dataset. (b) MuseGAN. (c) DuetHLSTM.

Figure 3: Visualization of average NCTM from three sources.

Furthermore, DuetHLSTM was able to learn the pitches used in the training data more effectively than the MuseGAN baseline. MuseGAN is missing half of the pitches in octave, while DuetHLSTM was able to learn to produce all 12 pitches in the octave, like the Lakh dataset, as can be seen in Figure 4. Its PCTM in Figure 5 is also more similar in distribution to the Lakh dataset.

To get a human-based assessment of the quality of our generated music, we conducted a user study with 13 human subjects. The subjects were university students who did not have prior knowledge of our project, though they were informed that the music was computer-generated. We asked the subjects to listen to 10 seconds each from two tracks of music generated by MuseGAN and by DuetHLSTM without being informed which was which, and rate the two samples on harmony, structure and rhythm, and whether the music sounded realistic. Our subjects were instructed to give ratings on the 1-5 scale, where 1 is poor and 5 is excellent. The average ratings on these three metrics are displayed in Table 2.

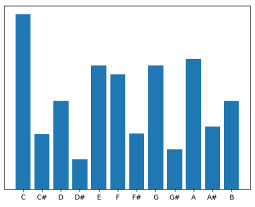
On average, subjects preferred our generated music over all the criteria, noting that the music generated by MuseGAN sounded interesting but sporadic, with one user comparing it to "cats on a keyboard." Multiple people also noted that the piano track of MuseGAN samples did not sound like they were physically possible to play. The subjects generally opined that music generated by DuetHLSTM seemed to have a real harmony and sounded natural. One subject commented that DuetHLSTM-generated music sounded similar to stock music, not particularly creative or innovative but generally inoffensive, "like elevator or mall music."

In addition to the objective metrics and human assessment, we have included visualizations of three samples each from the Lakh pianoroll dataset, results from the pretrained MuseGAN model available on GitHub, and samples generated after 30 epochs of training from DuetHLSTM in Figure 6.

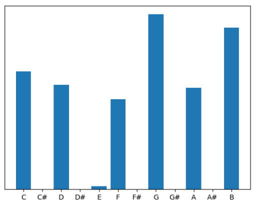
Note that, as our subjects mentioned, the MuseGAN results tend to be more noisy and sporadic, with many short notes being played for a brief period of time and over a wide range of pitches. We can also

Table 2: Average human rating of MuseGAN and DuetHLSTM.

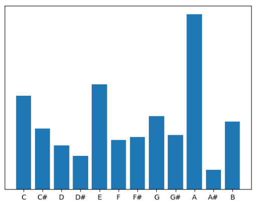
	Harmony	Structure/Rhythm	Realism
MuseGAN	2.23	2.23	2.08
DuetHLSTM	3.61	3.46	3.54



(a) Lakh dataset.

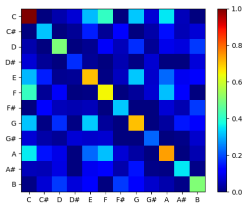


(b) MuseGAN.

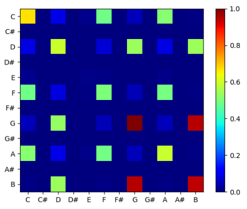


(c) DuetHLSTM.

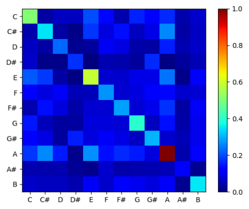
Figure 4: Visualization of PCH from three sources.



(a) Lakh dataset.



(b) MuseGAN.



(c) DuetHLSTM.

Figure 5: Visualization of average PCTM from three sources.

see that MuseGAN samples tend to have many more notes active at once than in the real music, both in the piano and strings tracks, which may in part explain the lower realism scores for MuseGAN.

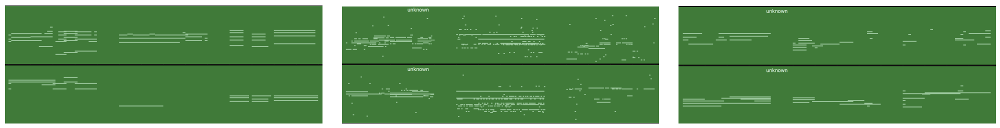
The DuetHLSTM pianorolls, on the other hand, look more similar to the ones in the Lakh dataset, with notes being held for similar lengths of time and in similar ranges. However, the DuetHLSTM samples tend to not have as much global structure as the real data, and are still noticeably more sporadic with respect to rhythm than music that was composed by humans.

6 Discussion and Analysis

We showed through human user study that our DuetHLSTM model has learned to generate pleasant music that is more similar to the training samples in the Lakh dataset than MuseGAN samples are. We can also see through the objective metrics, as well as in the visualizations of a few samples from each source, that the DuetHLSTM was able to more successfully learn the harmonic and rhythmic patterns in the Lakh training data.

Listening to the generated samples from DuetHLSTM, we note that our model tends to generate harmonic sets of notes, such as octaves and major chords, when multiple notes are played at the same time, more often than it generates dissonant sets. This seems to show that the model has learned what sounds good to the human ear, as dictated by the human-composed samples in the training set. A glance at the visualizations also shows that the accompaniment also often generates notes in temporal alignment with the melody, and listening to the notes reveals that these notes tend to be harmonizing to the melody notes (e.g. the same note, thirds, fifths, and octaves). This indicates that the hierarchical structure does, in fact, capture inter-track dependencies in both pitch and time. Finally, we note that the samples generated by DuetHLSTM, while similar to the input dataset, are distinct enough to disprove suspicions of "memorizing the input."

We do note a few weaknesses of our approach, however. Unlike human music composition, our model does not take into account any sort of bar structure, as we have flattened the bars into their constituent timesteps, and the training data does not indicate key signature or time signature. Though there may be inherent bar structure as well as key and time signatures in the training set, our model has no explicit indication of any of these, which makes it more difficult to capture an overall structure



(a) Lakh dataset. (b) MuseGAN. (c) DuetHLSTM.

Figure 6: Visualization two-track (piano and strings) pianorolls from three sources.

of the whole generated sequence. For comparison, the related works we have examined do not fully account for these structures and indications either.

Another shortcoming is the length of the sequences generated, since a sample of four bars of music is much shorter than conventional human compositions. We have not tried generating sequences longer than four bars, as our training data consisted of only four bars of music, but we suspect that longer musical sequences will need some way of maintaining an overall musical structure. Although DuetHLSTM does generate some recognizable patterns (ascending, descending, and combinations thereof), it does not tend to generate structured melody lines for the most part. This may be a shortcoming of the LSTM, as eventually we will lose some information about the past notes as time goes on. A possible solution is to incorporate some lookback features and skip connections, as in [1], or a shared structure generator, as in [3], but further experimentation is needed to see what would work best.

Lastly, we performed experiments on only two-track generated music. While DuetHLSTM should be able to be generalized to greater numbers of tracks, it would require modifications to the model architecture in order to take into account the additional tracks. In future work, we would be interested in investigating this aspect and determining if DuetHLSTM's success in maintaining inter- and intra-track dependencies can be generalized to more musical parts such as drums or a bassline.

In conclusion, in this paper, we presented a hierarchical LSTM architecture for generating multi-track polyphonic music. We showed that it was able to achieve better performance than the baseline according to a variety of metrics, both subjective and objective. However, generating pleasing and creative music is still a challenge that has much to be explored.

References

[1] Hang Chu, Raquel Urtasun, and Sanja Fidler. Song from pi: A musically plausible network for pop music generation. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017.

[2] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proc. of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[3] Hao-Wen Dong and Yi-Hsuan Yang. Convolutional generative adversarial networks with binary neurons for polyphonic music generation. In *Proc. Int. Society of Music Information Retrieval Conf (ISMIR)*, 2018.

[4] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2018.

[5] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. In *Proc. Int. Society of Music Information Retrieval Conf (ISMIR)*, 2017.

[6] Li-Chia Yang and Alexander Lerch. On the evaluation of generative models in music. In *Neural Computing and Applications*, Nov 2018.

[7] Haizi Yu and Lav R. Varshney. Towards deep interpretability (mus-rover ii): Learning hierarchical representations of tonal music. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017.