Generating Lead Sheets with Affect: A Novel Conditional seq2seq Framework

Dimos Makris
Information Systems, Technology, and Design
Singapore University of Technology and Design
Singapore
dimosthenis_makris@sutd.edu.sg

Kat R. Agres
Yong Siew Toh Conservatory of Music
National University of Singapore
Singapore
katagres@nus.edu.sg

Dorien Herremans
Information Systems, Technology, and Design
Singapore University of Technology and Design
Singapore
dorien_herremans@sutd.edu.sg

arXiv:2104.13056v1 [cs.SD] 27 Apr 2021

***Abstract*—The field of automatic music composition has seen great progress in the last few years, much of which can be attributed to advances in deep neural networks. There are numerous studies that present different strategies for generating sheet music from scratch. The inclusion of high-level musical characteristics (e.g., perceived emotional qualities), however, as conditions for controlling the generation output remains a challenge. In this paper, we present a novel approach for calculating the valence (the positivity or negativity of the perceived emotion) of a chord progression within a lead sheet, using pre-defined mood tags proposed by music experts. Based on this approach, we propose a novel strategy for conditional lead sheet generation that allows us to steer the music generation in terms of valence, phrasing, and time signature. Our approach is similar to a Neural Machine Translation (NMT) problem, as we include high-level conditions in the encoder part of the *sequence-to-sequence* architectures used (i.e., long-short term memory networks, and a Transformer network). We conducted experiments to thoroughly analyze these two architectures. The results show that the proposed strategy is able to generate lead sheets in a controllable manner, resulting in distributions of musical attributes similar to those of the training dataset. We also verified through a subjective listening test that our approach is effective in controlling the valence of a generated chord progression.**

***Index Terms*—Lead Sheet Generation, Emotion, Valence, seq2seq, Transformer**

I. Introduction

Developing computational music generation systems has been the focus of research for many years [1]–[4]. With the rapid development of deep generative models, their generation results have become hard to distinguish from real-world data in various applications. In the symbolic music generation domain, diverse strategies have been used for a variety of tasks [5]. Examples of music generation tasks are chorale harmonisation [6], multi-track generation using piano-rolls [7], [8] or lead sheets [9], and modifying a given piece of music with style transfer [10].

This work is funded by Singapore Ministry of Education Grant no. MOE2018-T2-2-161 and SRG ISTD 2017 129, as well as the RIE2020 Advanced Manufacturing and Engineering (AME) Programmatic Fund (No.A20G8b0102), Singapore.

In this work, we focus on generating lead sheets from scratch. A lead sheet is a form of musical notation that represents the fundamental elements of popular songs such as chords (using chord symbols), melody and sometimes lyrics. There has been some previous research on this particular task. De Boom et al. [11] proposed a two-stage generation system based on long-short term memory networks (LSTMs) to produce rhythm and chord events, for which a melody sequence is generated in a second (conditioned) phase. Liu & Yang [12] also introduced a two-stage generation system, which generates the lead sheet first, after which a polyphonic arrangement is produced as accompaniment using Generative Adversarial Networks (GANs).

Recently, there have also been research to let the user control the output of the generation by setting some constraints. These constraints are usually referred to as "high-level" musical parameters and may be relatively subjective, such as style and genre. Flow Composer [13] is an example of a conditional generative system that combines two Markov chains enriched by regular constraints, whereby the user sets the desired style of the lead sheet by selecting a corpus of existing lead sheets. [9] proposed an approach whereby structured lead sheets are generated, based on a mechanism (i.e., belief propagation) for efficiently sampling variations of existing musical sequences. In this system, the user can control certain parameters (e.g., similarity).

Extending this strategy of controlling the generation through high-level constraints imposed by the user, our system allows the user to control the emotion or valence of the generated music. Emotions and music are intrinsically connected [14], yet the qualities of the music that give rise to emotions are difficult to capture [15]. In order to create a training dataset, we first propose an approach for calculating the perceived emotions of the music from existing chord progressions. This allows us to label a training dataset of lead sheets with emotions, which we can then leverage to train the proposed conditional generative model.

Our approach for calculating emotions from chords is based on [16]–[18] who indicate that the mode or type (e.g., major, minor, seventh, etc.) of chords corresponds directly with the

valence of the music. High-level musical qualities such as emotions "suffer" from abstractness and subjectivity due to the fact that they require human annotations. In this work, however, instead of using human annotated labels, we filtered mood tags that correspond to different types of chords and use those to create a label for perceived emotions. The chosen tags were based on annotations by music experts [19]. In Section II we leverage this knowledge and propose a novel way of annotating chords with valence. To the best of our knowledge, there is no existing work that offers a method to manually calculate the valence of a chord progression.

We use the calculated valence as a high-level conditioning feature, along with others (e.g., time signature and grouping indicators inspired by [20]), in our proposed generative lead sheet system based on *sequence-to-sequence* architectures (LSTM [21] and Transformer [22]). Another novel aspect of our approach is a unique strategy to include the high-level user conditions as the encoder input, whereas the musical events of the lead sheet are predicted in the decoder. Thus, we approach the task of lead sheet generation much like a Neural Machine Translation (NMT) problem, except that we are translating 'conditions' into 'musical events'.

The remainder of this paper is organised as follows: Section II presents our proposed strategy for calculating the valence of chord progression within a lead sheet. Next, Section III shows the details of the proposed representation for conditional lead sheet generation. Sections IV & V detail the experimental setup and the evaluation of our approach, followed by a conclusion in Section VI.

II. Novel way to Measure Valence of Chords

A chord is defined as a set of two or more simultaneously played notes. A chord progression is a sequence of consecutive chords and often refers to the harmony of a song. It is considered to be a fundamental element of music that often influences the emotions a listener perceives [16]. This point becomes obvious when you listen to a new arrangement of a song in which the melody remains the same but the harmony is changed. The same piece can convey entirely different emotions if the chord progression has changed [17]. [23] mention the importance of certain chord types inside a progression for making the music sound "emotional". In addition, many studies show that major chords convey positive emotion and minor chords convey negative emotion (e.g., [24], [25]).

We will be representing the emotion of chords in terms of valence. Valence relates to the positivity or negativity of the emotion conveyed by a song and falls on a scale from positive ($+1$) to negative (-1) [26]. For instance, anger, fear, and sadness all have low (negative) valence. On the other hand, emotions such as happy, content, and joyful correspond to high (positive) valence. Given the fact that chord types have a direct impact on the valence of a piece [16], [17], we focus here on the effect that the chord progression of a lead sheet has on perceived valence. It is worth noting that 'arousal' (which refers to the energy level conveyed by the song) [26] is excluded in this approach due to the absence of tempo markings in our training dataset, given that arousal is typically strongly affected by tempo [27].

TABLE I
Chord types and their associated emotions, adapted from [19].

Chord Type (example)	Associated Emotions
Major (C)	Happiness, cheerfulness, confidence, brightness, satisfaction
Minor (Cm)	Sadness, darkness, sullenness, apprehension, melancholy, depression, mystery
Dominant Seventh (C7)	Funkiness, soulfulness, moderate edginess
Major Seventh (Cmaj7)	Romance, softness, jazziness, serenity, tranquillity, exhilaration
Minor Seventh (Cm7)	Mellowness, moodiness, jazziness
Dominant Ninth (C9)	Openness, optimism
Diminished (Cdim)	Fear, shock, spookiness, suspense
Suspended Fourth (Csus4)	Delightful tension
Seventh Minor Ninth (C7b9)	Creepiness, ominousness, fear, darkness
Added Ninth (Cadd9)	Steeliness, austerity

To the best of our knowledge, there is only one dataset in the symbolic domain that contains valence and arousal annotations: the VGMIDI dataset [28]. This dataset contains piano arrangements of 95 video game soundtracks in MIDI, annotated with valence and arousal values in the range of -1 to $+1$. Although the annotations are continuous, per beat, the input files are not close to the form of a lead sheet. There are also many occasions where harmony (i.e., chord progression) cannot be identified due to the unquantified nature of the midi files.

Thus, we propose a new method to **manually** calculate the valence of chords based on mood tags, as it is easier to find datasets with annotated mood tags, versus annotated valence and arousal. [19] found a relationship between modes of chords and associated evoked emotions from professional composers and musicians (see Table I). This list of chords with associated emotions has already shown to be useful in related research areas such us mood classification [29]. We leverage these findings and consider the modes and types of chords (major, minor, 7th, etc.) in our work, because this property has arguably the greatest influence on perceived mood [23].

We then use a mapping from Paltoglou and Thelwall [30] that matches Scherer's [31] emotion tags with corresponding valence and arousal values. This allow us to retrieve a valence value for the emotion tags for each chord type from Table I. A representation of this mapping is shown in Figure 1, where the valence and arousal values are represented on the x-axis and y-axis respectively (in the range of -1 to $+1$).

One issue we encountered when doing the mapping is that not all of the emotion tags from Table I were included in Scherer's model. To address this, we proceeded with two assumptions/modifications. First, the emotion tags that did not

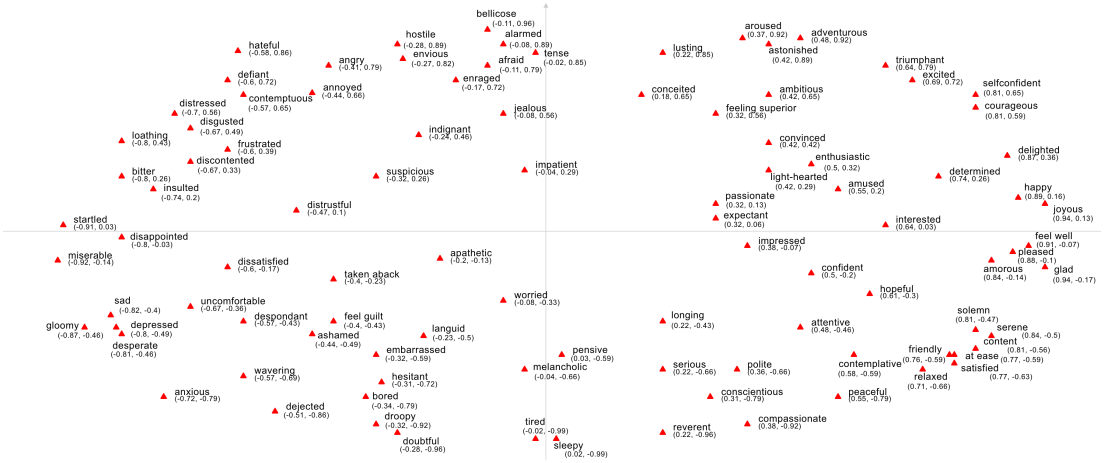

Fig. 1. Scherer [31]'s mapping of mood tags to the two-dimensional circumplex space model proposed by Russell [26]. The corresponding valence (x-axis) and arousal (y-axis) values are based on the coordinates provided by the mapping.

describe an "actual" emotion (e.g., jazziness) were removed. Next, the remaining tags that were not present in Scherer [31]'s model were matched as best as possible to synonyms that were present, with the help of a Music Psychologist who is a native English speaker. This resulted in a cleaned, reduced emotion tag list per chord type.

Using this method, we can find valence values based on chords by looking at the (cleaned) emotion tag list for each chord. The final valence value is the median valence of all the descriptive cleaned emotion tags. For instance, "Major" has five associated emotions. "Cheerfulness" was matched with "happiness" since they are synonyms, and "brightness" (which was not in [31]'s model) was mapped to "delighted". The extracted Valence values from Figure 1 are 0.89, 0.89, 0.51, 0.87 and 0.77 respectively, resulting in a final valence value of 0.87. This value makes intuitive sense because a major chord is often related to a positive mood.

III. Proposed Framework

In this section we propose a novel architecture for a "controllable", affective music generation system using high-level musical qualities imposed by the user. Because we consider the task of lead sheet generation to be a NMT task, we make use of popular *sequence-to-sequence* (seq2seq) architectures [21]. The user defines a sequence of musical attributes (conditions) in the *encoder* stage, which is then "translated" to a complete lead sheet in the *decoder* stage. Thus, we use two different input representations for both stages that are inspired by the event-based token representation from [32].

A. Encoder Representation - Controllable by the user

In our proposed approach, the **Encoder** takes a sequence of high-level musical parameters (conditions) as input, which allows the user to guide the generated music. The system allows for a high level of control, as the user can set the desired levels of the parameters for every bar. It is possible to have either varying or constant values of these parameters during generation. The following parameters are included:

- **Chord Valence**: During training, the valence for each chord type is calculated from the training set as described in Section II. To define the overall valence within a bar, we calculate the median for each chord inside that bar. Because valence ranges between -1 to 1, discretisation is needed. We use five discrete labels ("Low", "Moderate Low", "Neutral", "Moderate High" and "High" respectively) which are used to divide the valence range equally. Thus, chord types with close valence values will get the same descriptors.
- **Time Signature**: Symbols are used to describe the number of beats the "meter" of each bar of music [19]. Meter refers to the recurring pattern of accents that provide the pulse or beat of the music. Indicating and adjusting the time signature has a direct affect on the rhythm of the piece.
- **Grouping Indicators**: This feature contains extracted annotations which mark structurally coherent temporal regions of the music (i.e., different verses or refrains). We have five distinct symbols: two indicating the first two bars of a phrase, two for the last two bars and one more descriptor for the rest bars. These indicators allow the model to learn the initial and final events of a phrase, an approach that has been successfully used in the task of melodic harmonisation [33] and drums generation [20].

In addition to these high-level features, we implement a "low-level" feature called **event density** which is calculated as the number of events (either chord or melody) per bar. Note density will allow the user to control the number of generated events within a bar. The average number of events in the dataset used in our experimental setup (see Section IV-A) is 3.68 with a variance of 2.02 per bar. We use three discrete labels: one for low, medium and high note density with ranges of $[0-2]$, $[3-5]$, and $[6+]$ events, respectively.

Given the above described conditional input, the **Encoder** sequence can be defined as:

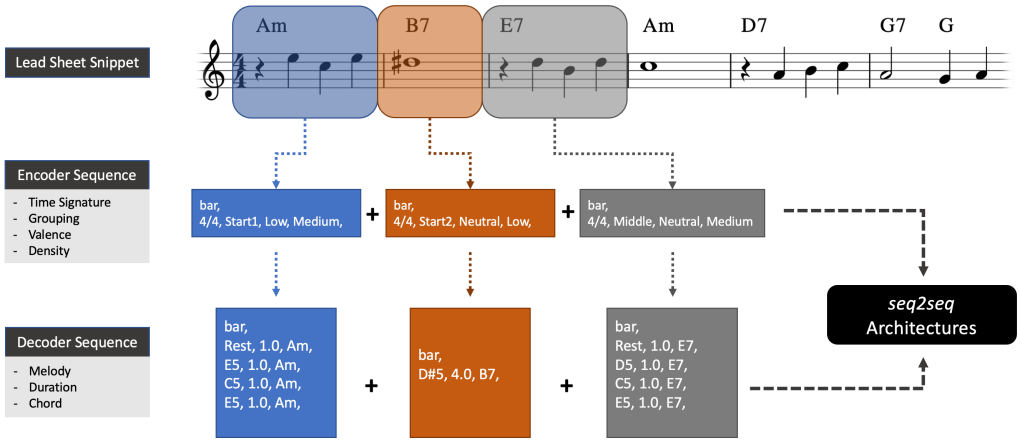

Fig. 2. Illustrated workflow of a lead sheet snippet transcribed to the proposed event representation for our encoder and decoder.

$$Enc_{seq} = (bar,\ h_1,\ bar,\ h_2,\ \ldots\ ,\ h_n)$$

whereby the *bar* event declares the start of a new bar, and h_i, is a vector of length 4 that represents the proposed high-level and low-level features for each bar i (with n being the total number of bars):

$$h_i = (TimeSig_i,\ Grouping_i,\ Valence_i,\ Density_i)$$

B. Decoder Representation - Lead Sheet Output

The **Decoder** outputs the generated sequence of lead sheet events per bar. This output can be formulated as follows:

$$Dec_{seq} = (bar,\ l_1,\ bar,\ l_2,\ \ldots\ ,\ l_n)$$

whereby the *bar* event declares a new bar, and l_i represents a series of lead sheet events (with n being the total number of bars defined in the Encoder stage) for the corresponding bar. Each lead sheet event consists of a chord symbol, melody pitch, and duration token. These tokens share the same dictionary that is built from the training dataset. The number of lead sheet events is variable for each bar, and can be controlled by the $Density_i$ feature in the Encoder. Therefore this can be formalised as:

$$l_i = (c_1,\ m_1,\ d_1,\ c_2,\ m_2,\ d_2,\ \ldots\ ,\ c_e,\ m_e,\ d_e)$$

where c, m and d represent a chord triad, melody pitch, and duration token respectively, with e indicating the generated number of events in the bar l_i. Figure 2 illustrates an example of a lead snippet transcribed to the proposed event representation for both Encoder and Decoder stages. This proposed representation is designed to be compatible with seq2seq architectures.

C. Model architectures

Our proposed Encoder-Decoder representation was designed to work with state-of-the-art seq2seq architectures that have been used successfully in the NMT field. We implemented the following two architectures:

1) **LSTM-based encoder-decoder**: Inspired by [11], we implemented a 3-layer Bidirectional LSTM (BiLSTM) [34] Encoder, consisting of 512 hidden units and a 3-layer Decoder of 1,024 hidden units with 30% dropout between consecutive layers. The BiLSTM states provided by the Encoder allow the lead sheet generator (i.e., Decoder) to look back as well as ahead at the sequence of music parameters defined by the user. It is worth noting that adding local or global attention mechanisms [35] to this network did not provide a better performance during training or generation. This may be due to the nature of the input representation of the encoder which is a sequence that describes a specific bar, whereas the decoder output is a list of multiple events within bars. Therefore, the size of the decoder is much larger than the encoder.
2) **Transformer**: We adapted the vanilla version of the transformer architecture from [22] in our proposed system. A total of 4 self-attention layers and 8 multi-head attention modules were used. The number of hidden units of the Feed-Forward layers was set to 1,536, with 20% dropout between consecutive layers.

D. Training and Generation

Due to the nature of our token-based encoding representation, we tackle the training and generation procedures with "teacher-forcing", a technique often used for NMT or text-generation tasks. Thus, it is worth noting that both seq2seq architectures have single outputs in the decoder stage, and hence generate a single token in every iteration (inspired by the encoding representations from [8], [32]) and not a triplet lead sheet event as described in the previous section. Therefore, in the decoder stage we have a single dictionary that includes all the tokens for chords, melody pitches, and durations.

We use the Adam optimizer with a learning rate of 0.001, and categorical cross-entropy as the loss function, for both

models. The batch size was set to 32 and the model was implemented using Tensorflow 2.x [36]. In order to control the diversity of the generation, we use a temperature τ (which was varied uniformly randomly between 0.8 and 1.2) to sample from the output distribution. Finally, in the generation stage, the user can either define the parameters manually for every bar, or randomly generate control parameter templates based on the statistics of the test set. The code and pre-processed dataset are available on GitHub[1].

IV. Experimental Setup

The goal of our proposed framework is to generate novel lead sheets that allow the user to control features such as valence at the bar-level. Our experiment focuses on the valence of the generated chord progressions, an approach made viable based on our unique strategy for chord valence calculation (Section II) that allows us to create a lead sheet dataset with labelled valence values. Therefore, we aim to evaluate:

1) Does the valence input by the user affect the perceived valence of the generated music as intended?
2) How effective is our proposed encoder/decoder representation and architecture to generate high quality, 'real' sounding music?

We address these research questions by both calculating an extensive set of evaluation metrics as well as a listening test described in Section IV-B2.

A. Data Collection and Pre-Processing

For our experiments, we use the Wikifonia dataset[2]. This dataset contains 6,675 lead sheets in MusicXML format, including diverse genres, from folk to popular music. After filtering out corrupted files that do not contain chord symbols or melody notation, we proceeded with the following pre-processing steps:

- All of the songs were transposed to the key of C major or A minor. Songs that contain key changes were split into different independent instances. In addition, there was a limit of up to 32 bars length for every song.
- Inspired by [11], we eliminated polyphonic melody parts and ignored ties between notes from different bars. Moreover, we unfolded repetitions since lead sheets can contain repeated phrases. Therefore if a repeat barline symbol occurs, we duplicate that particular phrase.
- We set restrictions on the available modes and chord types. Table II shows the permitted chord types to allow for our valence calculation method. Lead sheets including other chord types were removed. In addition, we removed inversions in the chord symbols.
- Multiple Time Signatures were detected in the original dataset, however, we only considered lead sheets with the 5 most common ones: $4/4$, $3/4$, $2/2$, $2/4$ and $6/8$.
- We only included lead sheets with the most frequent durations in the dataset, including triplets (of quarter notes or eighth notes). Finally, for melodic pitch, the permitted range was set from $G3$ (55) to $C6$ (84).

TABLE II
Occurrences of the different chord modes and their associated valence values in our dataset.

Chord Type	Valence	Occurrences
Major	0.87	333,232
Minor	-0.81	89,741
Dominant Seventh	-0.02	173,586
Major Seventh	0.83	19,617
Minor Seventh	-0.46	55,536
Dominant Ninth	0.51	12,944
Minor Ninth	-0.15	9,557
Diminished	-0.43	9,001

The resulting processed dataset contains 4,776 lead sheets with a variable length of 4 up to 32 bars. This was divided into training / validation / test sets with a ratio of 8:1:1. Both the Chord and Melody list of tokens include the "Rest" symbol.

B. Evaluation metrics

We conduct both a computational experiment and a user study to evaluate the quality of music generated by our proposed method. To evaluate how well our approach can create emotionally distinctive music based on the valence of the chord progression, we examine whether the participants in the listening study are able to identify the overall perceived emotions of the generated lead sheets.

1) Analytical measures: There is no standard way to quantitatively measure whether a lead sheet generation model has been trained well [37]. However, we adopt the following measures that were recently proposed in [7] and have been used to evaluate lead sheet generation from scratch [12].

- **Used Pitch Classes**: Average number of used pitch classes per bar for both melody and chord tracks.
- **Rest Events**: This is a modification of the proposed "Empty Bars" metric, as we do not encounter empty bars in our training dataset. Thus, this metric indicates the average ratio of rest events per bar for both melody and chord tracks.
- **Tonal Distance**: Measures the harmonicity between two given tracks [38]. Large values of Tonal Distance implies weaker inter-track harmonic relations between the Melody and the Chord track.

In addition, we propose two sets of calculated measures that can be used in the objective evaluation of generated lead sheets (see [39], [40]). First, by measuring the "compression ratio" of generated content we can measure the number of repeated patterns, which is related to "long-term structure". This can be calculated using the **Omnisia**[3] software which uses the COSIATEC compression greedy algorithm [41] and computes the following metrics:

- **Compression Ratio**: A measure of detecting repeated patterns such as themes and motives in the generated musical content.

[1]https://github.com/melkor169/LeadSheetGen_Valence

[2]Wikifonia archive is no more directly accessible from the web. You may contact the authors if you are interested in obtaining the original dataset.

[3]https://github.com/chromamorph/omnisia-recursia-rrt-mml-2019

TABLE III
RESULTS OF QUANTITATIVE EVALUATION IN TERMS OF THE PROPOSED METRICS (MEAN ± STANDARD DEVIATION).

	Used Pitch Classes		**Rest Events (%)**		**Tonal Distance**
	Melody	Chords	Melody	Chords	Melody - Chords
Training Dataset	2.5896 ± 1.1283	4.8602 ± 1.6168	0.0755 ± 0.1871	0.0132 ± 0.0574	1.4634
Proposed LSTM-based	**2.6503 ± 1.1166**	4.4447 ± 1.6251	**0.0806 ± 0.1993**	**0.0344 ± 0.1006**	1.6432
Proposed Transformer	2.3688 ± 1.0856	4.3101 ± 1.6335	0.0886 ± 0.2147	0.0424 ± 0.0884	**1.4918**
LSTM two stages [11]	2.2660 ± 1.1228	4.4483 ± 1.7735	0.1079 ± 0.2499	0.0591 ± 0.1223	1.5043
MuseGAN two tracks [7]	2.8575 ± 1.1643	**4.6541 ± 1.5931**	0.1695 ± 0.1895	0.0437 ± 0.0807	1.6543

	Pattern Metrics			**Tension Metrics**		
	Compression Ratio	Long Patterns (avg)	Short Patterns (avg)	Cloud Movement	Cloud Diameter	Distance to the Key
Training Dataset	1.7384 ± 0.1784	1.8039 ± 3.8745	15.3772 ± 5.9791	0.3197 ± 0.0987	2.4351 ± 0.3584	0.5639 ± 0.1083
Proposed LSTM-based	1.6599 ± 0.1113	0.8823 ± 1.8601	17.5720 ± 6.0227	0.3012 ± 0.0839	2.2780 ± 0.3426	0.5592 ± 0.1102
Proposed Transformer	**1.7654 ± 0.2185**	**2.1533 ± 3.9989**	**14.4458 ± 5.9062**	0.2742 ± 0.0994	2.2545 ± 0.3287	**0.5667 ± 0.1136**
LSTM two stages [11]	1.6715 ± 0.1267	0.8190 ± 1.8318	16.9420 ± 5.9048	**0.3168 ± 0.1006**	2.2266 ± 0.4118	0.6101 ± 0.1098
MuseGAN two tracks [7]	1.5355 ± 0.0664	0.2245 ± 1.0132	23.8170 ± 6.5830	0.2698 ± 0.2065	**2.4879 ± 0.6753**	0.6047 ± 0.1774

- **Average Long Patterns**: Measures the average number of the longest detected patterns (i.e., in terms of note events) within a lead sheet.
- **Average Short Patterns**: Indicates the average number of the shortest detected patterns.

Finally, we calculate tension measures proposed by [3], [42] to quantify the tension profile of a musical song. Musical tension forms an essential part of the experience of listening to music – increased tension levels can be subjectively described as "a feeling of rising intensity", while decreased tension is a "feeling of relaxation" [43]. We calculate the following measures which are based on the spiral array proposed by [44]:

- **Cloud Diameter**: Indicates the level of dissonance within a sliding window frame (i.e., "cloud").
- **Cloud Momentum**: Measures the distance (tonality movement) between different clouds
- **Tensile Strain**: Calculates the tonal distance between a cloud of notes and the key of the piece.

2) Listening test setup: We conducted an online listening test in which participants rated 15 short samples of lead sheets ranging from 20 to 40 seconds in duration. Each sample was presented in the form of a video clip which captures the playback of a lead sheet, so that the user could also see the chord symbols. The distribution of the samples was as follows: (i) 5 samples selected randomly from the test set, (ii) 5 samples generated with the Transformer architecture, and (iii) another 5 samples generated using the LSTM-based architecture.

First, we wanted to subjectively measure which proposed model sounds more "pleasant" and coherent. Thus, each participant was asked to rate each sample on a 5-point Likert scale, ranging from 1 (very low) to 5 (very high), using four criteria that were adapted from [7], [12]:

1) Rhythm: Whether the Rhythm events are pleasant.
2) Melody: How novel the generated Melody is.
3) Harmony: If the Chord Progression sounds coherent.
4) Naturalness: Whether the "humanised" element is perceived.

Second, in order to evaluate whether our proposed approach can really steer the valence of the generated chord progression, we asked the participants to rate their overall perceived valence of the chord progressions, using the five discrete labels from Section III-A which we refer to as valence descriptors in this experiment. For the test samples we calculated the average valence using our novel method presented in Section II. These valence values were then used as input conditions to generate music pieces. These new pieces were then rated by listeners in terms of valence. This allows us to evaluate whether the desired (input) valence is the same as the valence perceived by actual human listeners.

V. EXPERIMENTAL RESULTS

A. Quantitative Evaluation

We compare the evaluation scores of our proposed method with two related state-of-the-art approaches. Specifically:

- **LSTM - Two stages**: We re-created the two-stage LSTM model from [11] with the same configuration and hyper-parameters. In the first stage, Rhythm and Chord events are generated together using two stacked LSTM layers. Next, the previous output is fed to the BiLSTM layers to get the states and generate the Melody with two stacked LSTM layers again.
- **MuseGAN - Two tracks**: We adapted the model proposed by [12] which generates Lead Sheets from scratch as a first stage using the MuseGAN [7] architecture, a multi-track Sequential Generative Adversarial Network. We reduced the generated tracks to Melody and Chords only, and converted our training data to piano-roll format (an alternative symbolic representation) to be compatible with the network input.

Moreover since MuseGAN cannot generate sequences of variable length, we set a fixed length of 8 bars in a 4/4 Time Signature. Therefore, the training data was split into phrases in order to train all the models. We generated a total of 2,000 sequences. For our two proposed models (i.e., Transformer and

TABLE IV
LISTENING EXPERIMENT RATINGS (MEAN ± 95% CONFIDENCE INTERVAL) FOR PIECES GENERATED BY THE TWO PROPOSED ARCHITECTURES AS WELL AS EXISTING (HUMAN) COMPOSITIONS.

	Rhythm	**Melody**	**Chords**	**Naturalness**
Human Composer	**3.62 ± 0.16**	3.50 ± 0.17	3.56 ± 0.16	**3.64 ± 0.16**
LSTM-based	3.47 ± 0.17	3.43 ± 0.17	3.51 ± 0.15	3.28 ± 0.15
Transformer	3.53 ± 0.14	**3.68 ± 0.16**	**3.76 ± 0.14**	3.41 ± 0.17

LSTM-based), we used random sequences to act as Encoder inputs that were generated from normal distributions of the corresponding musical parameters which were acquired from the training dataset. Finally, the hyper-parameter temperature τ was fixed at 1.0 for all models.

Table III shows the results of all the proposed metrics for all the models. Values close to those extracted from the training data indicate that the generated fragments may have more chance to be musically valid as they match the properties of existing music. Regarding the first set of metrics proposed by [7], we can observe that the proposed LSTM-based seq2seq model achieves the best results in almost all categories except for the Tonal Distance and the average Used Pitch Classes for Chords. However, the small mean differences and high standard deviations in all models (even in the training set) suggest that strong conclusions cannot be made. One reasonable explanation for the high standard deviations may be the fact that Used Pitch Classes and Rest Events metrics are computed per bar in which the density of events can have large fluctuations.

Surprisingly, there is a huge difference in the pattern metrics. Based on these results, it seems that the Transformer is able to better generate sequences with long-term structure. This highlights the effectiveness of our proposed representation, which seems to work better in the Transformer architecture than the LSTM-based model. In addition, MuseGAN fails to produce repeated patterns, which can be explained by the small number of training instances and the nature of the piano-roll representation. Finally, regarding the tension measures, the two baseline models seem to have slightly better results but, once again, the difference is quite small, especially if we take into account the standard deviation.

B. Subjective Listening Test

A total of 42 subjects participated in our listening test. All of the participants indicated that they have a strong musical background and a profession related to the music industry (e.g., composers, producers, performers, and music information retrieval (MIR) researchers). We targeted these groups because we believe that our specialised experimental questions may have been confusing for users without sufficient musical knowledge. In an overall of 630 votes, Table IV reveals that both proposed models' ratings on the different aspects of musicality of the piece are very close to those given to the real compositions. In addition, the Transformer seems to generate "more coherent" and "more pleasant" compositions than the LSTM-based model, and may outperform the real compositions when it comes to melody and chords.

TABLE V
SUBJECTIVE EVALUATION OF THE ABILITY OF THE PROPOSED METHOD TO SUCCESSFULLY GENERATE A CHORD PROGRESSION ACCORDING TO THE CALCULATED VALENCE DESCRIPTOR. THE COLOUR OF THE USER VALENCE DESCRIPTOR INDICATES THE SUCCESS (GREEN) OR FAILURE (RED) OF THE MODEL, I.E. A MATCH BETWEEN THE INPUT CONDITION AND THE RESULTING USERS' AVERAGE RATING.

Track	**Calculated Valence Descriptor (input cond.)**	**User Valence Descriptor (of output)**	**Model**
1	Neutral	Neutral	Human Composer
2	Moderate High	Moderate High	Transformer
3	Neutral	Neutral	LSTM - based
4	Moderate High	Moderate High	Human Composer
5	Neutral	Moderate High	Transformer
6	Moderate Low	Moderate Low	LSTM - based
7	High	Moderate High	Human Composer
8	Moderate High	Moderate High	Transformer
9	Moderate Low	Moderate Low	LSTM - based
10	Neutral	Moderate Low	Human Composer
11	Moderate Low	Moderate Low	Transformer
12	High	Neutral	Human Composer
13	Moderate High	Moderate High	LSTM - based
14	Moderate Low	Moderate Low	Transformer
15	Moderate High	Moderate High	LSTM - based

The effectiveness of our proposed method to generate music with a particular valence is shown in Table V. This table shows the valence descriptors that were given as input conditions for generation (or, for the human pieces, those calculated manually), as well as the valence descriptors from averaged participant ratings in the experiment. From the 15 musical fragments in the experiment, 11 were matched correctly with the corresponding input valence descriptor. The 4 mismatches may be due to the level of subjectivity involved in labelling valence. Interestingly, we find that almost all mismatches are those on the real human compositions.

VI. CONCLUSIONS

This paper introduces a novel strategy for conditional lead sheet generation that allows the user to steer the music generation using high-level musical qualities. We present a novel approach for calculating the valence of a chord progression using pre-defined mood tags proposed by music experts. Then, by tackling the task of lead sheet generation as a Neural Machine Translation problem, we propose a new approach to represent musical conditions such as valence as input to the *encoder* stage of popular *sequence-to-sequence* architectures. These conditions are then translated into musical lead sheet events in the *decoder* stage. An analytical experiment and listening test show that the proposed strategy is able to produce lead sheets in a controllable manner with similar musical attributes to the training dataset, that contain long-term structure, and which sound coherent and pleasant. In addition, the results of the listening test indicate the effectiveness of the proposed strategy to calculate and control the valence of a generated

chord progression. In future research, we might examine the effect of using a noise-robust loss function (e.g. [45]) on the model performance, along with adding more high-and low-level musical conditions such us arousal and just-intonation.

REFERENCES

[1] L. A. Hiller Jr and L. M. Isaacson, "Musical composition with a high speed digital computer," in *Audio Engineering Society Convention 9*. Audio Engineering Society, 1957.

[2] D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," *ACM Computing Surveys*, vol. 50, no. 5, pp. 1–30, 2017.

[3] D. Herremans and E. Chew, "Morpheus: generating structured music with constrained patterns and tension," *IEEE Transactions on Affective Computing*, vol. 10, no. 4, pp. 510–523, 2017.

[4] I. Deliège and G. A. Wiggins, *Musical creativity: Multidisciplinary research in theory and practice*. Psychology Press, 2006.

[5] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep learning techniques for music generation*. Springer, 2020.

[6] G. Hadjeres, F. Pachet, and F. Nielsen, "Deepbach: a steerable model for bach chorales generation," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1362–1371.

[7] 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 *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[8] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *International Conference on Learning Representations*, 2018.

[9] F. Pachet, A. Papadopoulos, and P. Roy, "Sampling variations of sequences for structured music generation." in *ISMIR*, 2017, pp. 167–173.

[10] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer," *arXiv:1809.07600*, 2018.

[11] C. De Boom, S. Van Laere, T. Verbelen, and B. Dhoedt, "Rhythm, chord and melody generation for lead sheets using recurrent neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 454–461.

[12] H.-M. Liu and Y.-H. Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 722–727.

[13] A. Papadopoulos, P. Roy, and F. Pachet, "Assisted lead sheet composition using flowcomposer," in *International Conference on Principles and Practice of Constraint Programming*. Springer, 2016, pp. 769–785.

[14] L. B. Meyer, *Emotion and meaning in music*. University of chicago Press, 2008.

[15] K. W. Cheuk, Y.-J. Luo, B. Balamurali, G. Roig, and D. Herremans, "Regression-based music emotion prediction using triplet neural networks," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[16] P. N. Juslin and J. Sloboda, *Handbook of music and emotion: Theory, research, applications*. Oxford University Press, 2011.

[17] Y.-H. Cho, H. Lim, D.-W. Kim, and I.-K. Lee, "Music emotion recognition using chord progressions," in *2016 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016, pp. 002 588–002 593.

[18] K. Zhao, S. Li, J. Cai, H. Wang, and J. Wang, "An emotional symbolic music generation system based on lstm networks," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. IEEE, 2019, pp. 2039–2043.

[19] W. Chase, *How music really works!: the essential handbook for songwriters, performers, and music students*. Roedy Black Pub., 2006.

[20] D. Makris, M. Kaliakatsos-Papakostas, I. Karydis, and K. L. Kermanidis, "Conditional neural sequence learners for generating drums' rhythms," *Neural Comput. and Appl.*, vol. 31, no. 6, pp. 1793–1804, 2019.

[21] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv:1406.1078*, 2014.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[23] I. Lahdelma and T. Eerola, "Single chords convey distinct emotional qualities to both naïve and expert listeners," *Psychology of Music*, vol. 44, no. 1, pp. 37–54, 2016.

[24] D. R. Bakker and F. H. Martin, "Musical chords and emotion: Major and minor triads are processed for emotion," *Cognitive, Affective, & Behavioral Neuroscience*, vol. 15, no. 1, pp. 15–31, 2015.

[25] K. J. Pallesen, E. Brattico, C. Bailey, A. Korvenoja, J. Koivisto, A. Gjedde, and S. Carlson, "Emotion processing of major, minor, and dissonant chords: a functional magnetic resonance imaging study," *Ann. N. Y. Acad. Sci.*, vol. 1060, no. 1, pp. 450–453, 2005.

[26] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[27] E. Coutinho and A. Cangelosi, "Musical emotions: Predicting second-by-second subjective feelings of emotion from low-level psychoacoustic features and physiological measurements." *Emotion*, vol. 11, no. 4, p. 921, 2011.

[28] L. Ferreira and J. Whitehead, "Learning to generate music with sentiment." in *ISMIR*, 2019, pp. 384–390.

[29] B. Schuller, J. Dorfner, and G. Rigoll, "Determination of nonprototypical valence and arousal in popular music: features and performances," *EURASIP J. Audio, Speech Music. Process.*, vol. 2010, pp. 1–19, 2010.

[30] G. Paltoglou and M. Thelwall, "Seeing stars of valence and arousal in blog posts," *IEEE Transactions on Affective Computing*, vol. 4, no. 1, pp. 116–123, 2012.

[31] K. R. Scherer, "What are emotions? and how can they be measured?" *Social science information*, vol. 44, no. 4, pp. 695–729, 2005.

[32] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Comput. Appl.*, vol. 32, no. 4, pp. 955–967, 2020.

[33] C. Tsougras, D. Makris, E. Cambouropoulos, and M. Kaliakatsos-Papakostas, "Learning and creating novel harmonies in diverse musical idioms: An adaptive modular melodic harmonisation system," *Journal of Creative Music Systems*, vol. 1, no. 1, 2016.

[34] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[35] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[37] K. Agres, J. Forth, and G. A. Wiggins, "Evaluation of musical creativity and musical metacreation systems," *Computers in Entertainment (CIE)*, vol. 14, no. 3, pp. 1–33, 2016.

[38] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, 2006, pp. 21–26.

[39] C.-H. Chuan and D. Herremans, "Modeling temporal tonal relations in polyphonic music through deep networks with a novel image-based representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[40] G. Zixun, D. Makris, and D. Herremans, "Hierarchical recurrent neural networks for conditional melody generation with long-term structure," in *Proc. of the International Joint Conference on Neural Networks (IJCNN)*, 2021.

[41] D. Meredith, "Cosiatec and siateccompress: Pattern discovery by geometric compression," in *Proc. ISMIR*, 2013.

[42] D. Herremans, E. Chew *et al.*, "Tension ribbons: Quantifying and visualising tonal tension." in *Proc. of Int. Conf. on Technologies for Music Notation and Representation (TENOR)*, vol. 2, Cambridge, UK, 2016, pp. 8–18.

[43] M. M. Farbood, "A parametric, temporal model of musical tension," *Music Perception*, vol. 29, no. 4, pp. 387–428, 2012.

[44] E. Chew, "The spiral array: An algorithm for determining key boundaries," in *Int. Conf. on Music and Artificial Intelligence*. Springer, 2002, pp. 18–31.

[45] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *arXiv:1805.07836*, 2018.