IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 19, 2021, accepted March 8, 2021, date of publication March 12, 2021, date of current version March 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3065831

Chord Conditioned Melody Generation With Transformer Based Decoders

KYOYUN CHOI, JONGGWON PARK, WAN HEO, SUNGWOOK JEON, AND JONGHUN PARK

Department of Industrial Engineering, Seoul National University, Seoul 08826, Republic of Korea
Institute for Industrial Systems Innovation, Seoul National University, Seoul 08826, Republic of Korea

Corresponding author: Jonghun Park (jonghun@snu.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant through the Ministry of Science and ICT (MSIT) under Grant NRF-2019R1F1A1053366, and in part by the MSIT and National IT Industry Promotion Agency (NIPA)'s HPC Support Project.

ABSTRACT For successful artificial music composition, chords and melody must be aligned well. Yet, chord conditioned melody generation remains a challenging task mainly due to its multimodality. While few studies have focused on this task, they face difficulties in generating dynamic rhythm patterns aligned appropriately with a given chord progression. In this paper, we propose a chord conditioned melody Transformer, a K-POP melody generation model, which separately produces rhythm and pitch conditioned on a chord progression. The model is trained in two phases. A rhythm decoder (RD) is trained first, and subsequently a pitch decoder is trained by utilizing the pre-trained RD. Experimental results show that reusing RD at the pitch decoding stage and training with pitch varied rhythm data improve the performance. It was also observed that the samples produced by the model well reflected the key characteristics of dataset in terms of both pitch and rhythm related features, including chord tone ratio and rhythm distribution. Qualitative analysis reveals the model's capability of generating various melodies in accordance with a given chord progression, as well as the presence of repetitions and variations within the generated melodies. With subjective human listening test, we come to a conclusion that the model was able to successfully produce new melodies that sound pleasant in terms of both rhythm and pitch (Source code available at https://github.com/ckycky3/CMT-pytorch).

INDEX TERMS Attention mechanism, computer generated music, deep learning, melody generation, neural networks.

I. INTRODUCTION

With the rapid development of machine learning techniques, research applying them to numerous music information retrieval tasks [1]–[4] can often be found. In particular, the subject of music generation employing deep neural networks has been explored in diverse ways: creating various forms of music including piano music [5], [6], lead sheet [7], and multitrack MIDI [8]–[10], or modifying a given piece of music with style transfer [11] or reinforcement learning [12], to name a few.

In order for a piece of music to be pleasant to listen to, chords and melody must be in harmony. It is essential to capture the relationship between chords and melody in various music composition tasks. Yet, chord conditioned melody generation is challenging mainly due to its large search space and the absence of standard quantitative measures for performance assessment. Most previous work had limitations in terms of expressiveness of chords or suffered from generating dynamic rhythm patterns adequately aligned with a given chord progression.

Since Transformer [13] and its extensions showed impressive results in many sequence modelling tasks [14], [15], there have been several studies that have adopted the concept of attention in the domain of music information retrieval [16]–[18]. This paper focuses on the task of generating monophonic melody for K-POP music, conditioned on a given chord progression. We introduce a novel chord conditioned melody generation model, named CMT (Chord conditioned Melody Transformer), and a training method for the model so that it can generate a melody with appropriate rhythms for a given chord progression.

The training procedure consists of two phases, and rhythm and pitch decoders are jointly trained at both phases. At the

The associate editor coordinating the review of this manuscript and approving it for publication was Bin Liu.

first phase, training data are augmented through pitch shifts to make the rhythm decoder robust against chord variations. At the second phase, the rhythm decoder is retained from the first phase and fine tuned, while the pitch decoder is newly initialized. In particular, when producing melodies with the trained model, rhythm generation precedes the generation of each note's pitch, as the pitch decoder utilizes the rhythm decoder's intermediate rhythm representation.

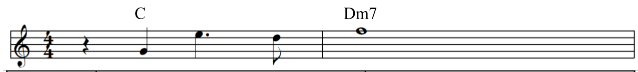
Experimental results show that the proposed two-phase training approach as well as the data augmentation improve the model's performance. Quantitative analyses were conducted against several metrics to experimentally show that generated melodies are in harmony with given chord progressions and have characteristics similar to those of dataset. It turns out that generated melodies adaptively follow chord progressions, showing repetitions and variations. Furthermore, subjective listening test demonstrates that the proposed model produces better melodies than the alternative model considered.

Bar		1								2							
Chord (12dim.)	B																
	A#																
	A									1	1	1	1	1	1	1	1
	G#																
	G			1	1	1	1	1	1								
	F#																
	F									1	1	1	1	1	1	1	1
	E			1	1	1	1	1	1								
	D#																
	D									1	1	1	1	1	1	1	1
	C#																
	C			1	1	1	1	1	1	1	1	1	1	1	1	1	1
Rhythm		R	R	O	H	O	H	H	O	O	H	H	H	H	H	H	H
Pitch		R	R	G4	H	E5	H	H	D5	F5	H	H	H	H	H	H	H

FIGURE 1. An example of chord, rhythm and pitch representations with the first 2 bars of the song "I Have A Dream" from the movie "Mamma Mia!". O, H and R stand for onset, hold, and rest, respectively.

II. RELATED WORK

A. CHORD CONDITIONED MUSIC GENERATION

Since chord and melody are two of the most essential elements that make up modern pop music, many studies have been conducted to capture their relationships in music generation. While there have been various research results on arranging a chord progression for melody [19]–[22], the task of chord conditioned melody generation has been relatively less investigated.

Brunner *et al.* [23] presented JamBot that consists of two LSTMs (Long Short-Term Memory) [24] for generating chord progressions and chord conditioned polyphonic music, respectively. Predicted chords and piano-rolls were extracted from MIDI data to respectively train chord and polyphonic LSTMs. The heuristic applied to extract chords assumed that the three most played notes of a bar are the notes that make up the triad chord of the bar. This heuristic has drawbacks in that the inferred chords are inaccurate and limited to only triads. Furthermore, only the 50 most frequent chords were used for training the chord LSTM, and the model was unable to capture the relationship between different chords since chords were symbolized as one-hot vectors.

MIDINet [25] explored a GAN (Generative Adversarial Network) [26] framework to generate multitrack music in three ways: from scratch, with chords, or with a priming melody. With regard to the chord conditioned generation, a CNN (Convolutional Neural Network) generator takes random noises with additional chord vectors to produce piano-roll matrices. Each chord of a bar is represented as a chord vector with 13 dimensions: 12 dimensional one-hot vector for a root note and an additional dimension to specify a chord type, either major or minor. While such chord vectors maintain the relationship between chords with respect to a root note and a chord type, they can only express 24 triads.

Chord based rhythm and melody cross-generation model (CRMCG) introduced by Zhu *et al.* [27] aims to build rhythms and melodies separately. It is composed of a chord GRU (Gated Recurrent Unit) [28] and two auto-encoders, one for rhythm and the other for melody. Both rhythm and melody auto-encoders utilize GRUs for encoding and decoding. Chords given every 2 bars pass through the chord GRU. Together with the result of melody encoder, outputs of rhythm encoder and chord GRU are then fed into the rhythm and melody decoders, respectively. The model's downside compared to this work is that the chord vectors are not fed into the rhythm decoder, not reflecting the fact that the rhythms used for chords are closely related to that of melodies.

Another attempt to disentangle rhythms and pitch classes was made by Yang *et al.* [29]. Explicitly-constrained conditional variational auto-encoder (EC^2-VAE) extracts rhythm and pitch latent variables from chord and melody. A rhythm decoder estimates distributions over rhythm tokens from a latent rhythm variable, and a global decoder receives the distributions to reconstruct melody. If the rhythm distributions have been trained so that the probability of repeating preceding rhythm patterns is high, generated melody would simply repeat some rhythm pattern over and over again. Contrary to EC^2-VAE, the pitch decoder of CMT takes as input the rhythm tokens sampled from the distributions of the rhythm decoder. In this way, other rhythm tokens with lower probabilities can also be sampled even if the rhythm distributions were trained to repeat preceding rhythm patterns, leading to more variations in rhythm.

B. ATTENTION MECHANISM AND MUSIC TRANSFORMER

The purpose of attention mechanism in tasks such as machine translation is to compute how much weights should be multiplied to each element of a source sequence when generating a target sequence. The situation in which the source sequence is equal to the target sequence is called self-attention. Transformer [13] is a sequence model that relies solely on the self-attention and it showed outstanding performance in machine translation. It has an encoder-decoder structure, utilizing the self-attention for both encoder and decoder. The

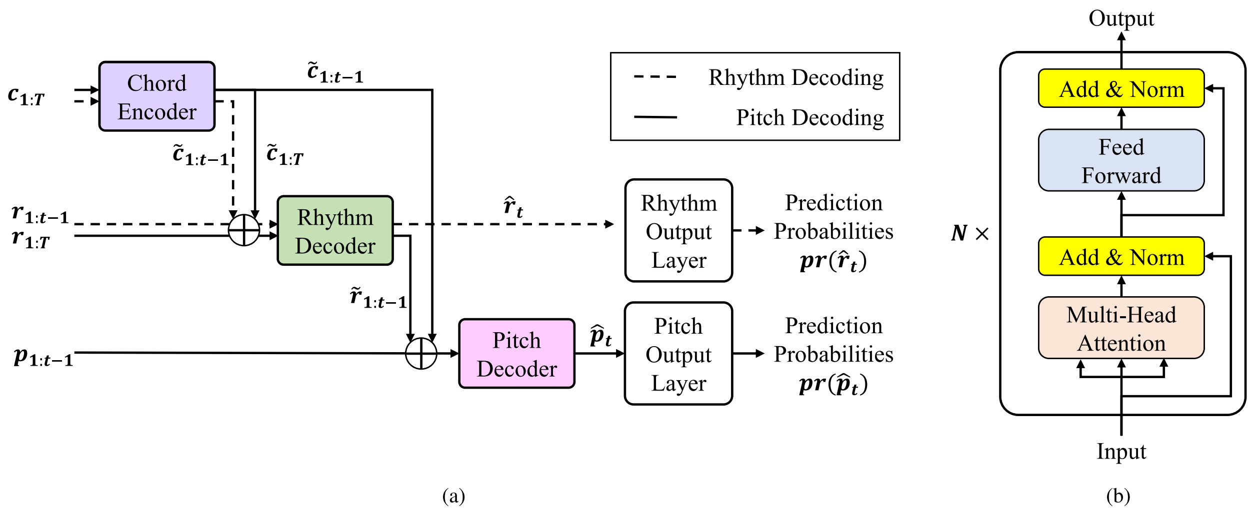


FIGURE 2. Structure of CMT (Chord conditioned Melody Transformer). (a) Flow diagram for the rhythm decoding and pitch decoding, which are depicted as dotted lines and solid lines, respectively. ⊕ denotes concatenation and c, r, p stand for chord, rhythm, and pitch, respectively. (b) Detailed architecture of the rhythm and pitch decoders. N identical self-attention blocks are stacked.

decoder needs an upper triangular mask to prevent attending to succeeding elements, whereas no such masking is needed in the encoder since the whole source sequence is available.

However, depending only on the self-attention without convolution or recurrence results in loss of information about the order in a sequence. To overcome this, Transformer injects absolute positional encoding into the input. Shaw *et al.* [30] suggested relative self-attention that models relative distance between elements more explicitly. Unfortunately, since relative distance between every element pair needs to be computed, memory complexity grows quadratically along with the sequence length.

Huang *et al.* [16] came up with a memory efficient relative self-attention model so that it can be employed for much longer sequences. As an application, the authors presented Music Transformer, a music generation model that can capture long-term structures. It demonstrated state-of-the-art performance in the task of piano music generation. Music Transformer adopts Transformer's decoder, except the fact that the attention to the encoder's output is omitted since there is no encoder. We employ the same Transformer architecture in our rhythm and pitch decoders.

III. METHODS

A. DATA REPRESENTATION

Representations of chord and rhythm in this paper are identical to those of [29]. Chords are symbolized by 12 dimensional binary vectors. Each component of chord vector, c, corresponds to an activated pitch class of a chord. On the other hand, three dimensional one-hot rhythm vector r indicates one of three types of rhythmic elements: onset of a note, holding state of an onset note, and rest.

For pitch representation, we employed a 50 dimensional one-hot pitch vector, denoted as p. The first 48 dimensions respectively indicate the onset of MIDI pitch from 48 (C3) to 95 (B6), and two additional dimensions signify holding state and rest, respectively. Any data out of the pitch range are shifted in octaves to fit within the range.

Figure 1 illustrates an example of the data representations with the first 2 bars of the song "I Have A Dream". For the sake of simplicity, unit time step in the example is assumed to be the length of the 8th note in Figure 1, which differs from our actual implementation of the minimum time length of the 16th note.

B. MODEL ARCHITECTURE

Architecture of the model proposed in this work, CMT, is shown in Figure 2. CMT consists of three modules: chord encoder, rhythm decoder, and pitch decoder, denoted as $\mathbf{E_c}$, $\mathbf{D_r}$, and $\mathbf{D_p}$ in the following definitions, respectively.

Let T be the total number of time steps. The goal of CMT is to generate rhythm sequence $r_{1:T} = \{r_1, \ldots, r_T\}$ and pitch sequence $p_{1:T} = \{p_1, \ldots, p_T\}$ auto-regressively from a given chord sequence $c_{1:T} = \{c_1, \ldots, c_T\}$, where subscript indicates the index of a sequence element. Embedding matrices of chord, rhythm, and pitch, denoted as M_c, M_r, and M_p, respectively, are multiplied to their corresponding vectors before the vectors are fed into the modules.

1) CHORD ENCODER

The chord encoder (CE) takes the form of a bidirectional LSTM (BLSTM) [31]. Replacing the self-attention encoder in the Transformer with a BLSTM encoder led to reduction in the fluctuation of loss values. CE's output $\tilde{c}_{1:T}$ can be formulated as follows:

$$\tilde{c}_{1:T} = \mathbf{E_c}(M_c \cdot c_{1:T}) \tag{1}$$

2) RHYTHM DECODER

The rhythm decoder (RD) consists of a stack of N self-attention blocks, as depicted in Figure 2(b). When decoding the t-th rhythm token for $2 \leq t \leq T$, only the preceding rhythm sequence $\{r_1, \ldots, r_{t-1}\}$ is accessible and the remaining $T-(t-1)$ tokens must be masked. The masked rhythm sequence is then given as an input to RD, together with $\tilde{c}_{1:t-1}$, to yield $\hat{r}_t$:

$$\hat{r}_t = [\mathbf{D_r}((M_r \cdot r_{1:t-1}) \oplus \tilde{c}_{1:t-1})]_t \tag{2}$$

where $\oplus$ denotes concatenation and subscript '$1 : t-1$' indicates a sequence of tokens or vectors of length T with $T-(t-1)$ masks at the end. Note that since the output of a BLSTM contains hidden states from both forward and backward directions, each non-masked vector in $\tilde{c}_{1:t-1}$ is produced considering the whole chord sequence $c_{1:T}$. An output layer that consists of a fully-connected layer and a softmax layer, is added to convert $\hat{r}_t$ into probability distributions over rhythm tokens, $pr(\hat{r}_t)$.

3) PITCH DECODER

The pitch decoder (PD) consists of another stack of N self-attention blocks. At the decoding step of the t-th pitch token, the whole sequences of both chord and rhythm, $c_{1:T}$ and $r_{1:T}$ respectively, are available. Rather than training another separate rhythm encoder, we reuse the intermediate rhythm representation $\tilde{r}_{1:T}$ from RD:

$$\tilde{r}_{1:T} = \mathbf{D_r}((M_r \cdot r_{1:T}) \oplus \tilde{c}_{1:T}) \tag{3}$$

After obtaining $\tilde{r}_{1:t-1}$ from $\tilde{r}_{1:T}$, together with $\tilde{c}_{1:t-1}$ and $\tilde{r}_{1:t-1}$, PD receives a masked sequence $p_{1:t-1}$ of length T, which consists of $(t-1)$ preceding pitch tokens and $T-(t-1)$ masks. The output $\hat{p}_t$ can be formulated as follows:

$$\hat{p}_t = [\mathbf{D_p}((M_p \cdot p_{1:t-1}) \oplus \tilde{c}_{1:t-1} \oplus \tilde{r}_{1:t-1})]_t \tag{4}$$

$\hat{p}_t$ is also converted into probability distributions over pitch tokens $pr(\hat{p}_t)$ after passing through another fully-connected layer, followed by a softmax layer.

C. TWO-PHASE TRAINING

There are two loss terms for CMT, namely rhythm loss and pitch loss. The training procedure for CMT is divided into two phases. The first phase trains RD, and then PD is trained at the second phase with the pre-trained RD. Since computing $\hat{p}_t$ requires $\tilde{r}_{1:T}$, RD's parameters are involved in minimization of the pitch loss too. Since back propagating only the pitch loss at the second phase might result in performance degradation of the pre-trained RD, the sum of rhythm loss and pitch loss is minimized at the second phase while RD and PD are jointly trained as in [32], [33].

D. PITCH VARIED RHYTHM DATA

Since distribution of pitch classes depends on the key, different keys in different training data may disrupt the training of PD. There are two ways to overcome this difficulty: shifting the pitch of melodies as well as chords by semitones from -5 to $+6$, resulting in 12 times more data, or shifting every song into the same key. While being equivalent in the number of original songs the model is trained with, the latter is much more efficient in terms of time and computing resources. Therefore, all the training data have been shifted into one key, C for major and A for minor, respectively in this paper. PD is trained to generate a melody in C major or A minor, and melody in any other key can be produced by shifting the result up or down by certain semitones.

On the other hand, as for rhythm, pitch varied dataset is more valuable than the single key dataset when training RD. Melodies in 12 different keys obtained by shifting the pitch of one melody have same rhythm but differ only in pitch, and the same is true for the chords. RD, which receives only the chords as input, can be trained with 12 times more instances that has the same starting time and duration of chords, yielding the same ground truth rhythm labels but with different chords. By training RD with the pitch varied dataset, we anticipate that RD will not only be trained to capture the timing of chords but also be robust to their pitch classes.

IV. EXPERIMENTS

A. DATA

1) DATASET

All datasets for quantitative experiments in this work came from EWLD (Enhanced Wikifonia Leadsheet Dataset) [34]. It is a music leadsheet dataset of more than 5,000 scores. After filtering out inappropriate genres for singing melody such as traditional, piano, chorale, and scores that do not contain chord or melody, approximately 4,000 scores were divided into training / validation / test sets by ratios of 8:1:1.

For qualitative evaluation, models were trained on a custom K-POP score dataset. About 1,400 leadsheets of K-POP melodies were acquired from a commercial sheetmusic website and converted to MusicXML by use of commercial leadsheet recognition software. The ratios of training, validation, test split were 8:1:1, respectively.

2) PREPROCESSING

All the songs were shifted to C major or A minor key. Songs that contain key changes were split into multiple songs with a single key each. Each of these splitted songs were considered as an independent song, except that they were grouped together to be included in the same set when dividing the data into training / validation / test sets.

Each song in the dataset was further divided into pieces of music with 8 bars, with a sliding window of 4 bars. The unit note considered was 16th note. As a result, there were 128 unit notes in each data instance. Moreover, data instances were discarded if the melody does not satisfy any of the following conditions: MIDI pitch range is limited to 4 octaves based on the assumption that the range of human singing voice would not exceed 4 octaves. For similar reasons, two consecutive notes should not be more than one octave apart in pitch. The percentage of the rest should be less than 25%, in other words, 32 time steps.

B. TRAINING AND GENERATION

Negative log-likelihood was employed as a loss function for training of RD. For PD, focal loss [35] was employed to resolve the label imbalance problem since there were much more tokens of holding state and rest than those of pitch onset.

Let $pr(r_t)$ and $pr(p_t)$ respectively be the output probabilities of RD and PD for the ground truth labels. The total loss $L = L_r + L_p$ is to be minimized where rhythm loss L_r and pitch loss L_p are respectively defined as follows:

$$L_r = -\sum_{t=2}^{T} \log\left(pr(r_t)\right)$$
$$L_p = -\sum_{t=2}^{T} (1 - pr(p_t))^{\gamma} \log\left(pr(p_t)\right) \tag{5}$$

with focusing parameter $\gamma = 2$.

Our model was implemented with PyTorch. The embedding dimensions of chord, rhythm, and pitch were 128, 32, and 256, respectively. A single-layer BLSTM with the hidden dimension of 56 was employed as CE. For the rhythm and pitch decoders, 8 ($=N$) self-attention blocks of 16 heads were stacked with dropout probability of 0.2. To enable stacking identical self-attention blocks and concatenating embedding vectors to modules' outputs, the hidden dimensions of RD and PD were respectively set to be 144 and 512. Adam optimizer [36] was adopted as an optimizer. The initial learning rate was 10^{-4}, decaying with the factor 0.5 if the validation loss does not decrease for more than 4 epochs, until the minimum value of 10^{-6}.

At generation stage, rhythm and pitch sequences were decoded auto-regressively, implying that the t-th element of a sequence was generated after the preceding $(t-1)$ elements had been generated. Whereas ground truth rhythm sequence was given as input when training PD, rhythm sequence generated by RD was fed instead during the melody generation by PD. To generate melodies with dynamic rhythms rather than with simple repetitions of a single pattern, rhythm tokens were sampled with probability $pr(\hat{r}_t)$ at each time step t instead of applying argmax. Pitch tokens were sampled with top-5 sampling strategy for the same reason. At each time step, a pitch token was sampled from one of the 5 most plausible tokens based on the rescaled top-5 probabilities.

Since CMT is a variant of Transformer, it requires a seed to generate a melody. To start constructing melody of 8 bars, the first token was chosen heuristically depending on a given chord sequence as follows. If there was no chord at the first time step, the rest token was set as the first token for both rhythm and pitch. Otherwise, the onset of the pitch corresponding to the starting chord's root note was set as the first token.

C. EXPERIMENT SETTINGS

For CMT, five experiments with different settings were conducted. The most basic setting was 1 phase (1P) training, while the training procedure of the other settings was composed of 2 phases (2P). At the first phase of 2P settings, the effects of pitch varied rhythm data and the loss terms were examined. We use the following abbreviation scheme for referring to the settings: PV for training with the pitch varied data and SK for training with the single key data. With regard to the loss term, rhythm only (RO) settings and rhythm with pitch (RP) settings were compared. As a result, the five experiment settings considered were 1P, 2P-PV-RO, 2P-SK-RO, 2P-PV-RP, and 2P-SK-RP, respectively.

During the first phase, CMT was trained with the pitch varied rhythm data for PV settings while the single key data were used for the training of SK settings. To ensure that the model is trained with equivalent amount of data for both settings, the number of maximum training epochs was 100 for PV settings and 1,200 for SK settings. Only the rhythm loss L_r was minimized for RO settings, while the loss term for RP settings was $L = L_r + L_p$.

For the second phase of all 2P settings, the model was trained with the single key data for 100 epochs. At the beginning of the second phase, RD's parameters trained from the first phase were restored while the parameters of CE and PD were initialized randomly. The learning rate of RD's parameters was fixed to 10^{-6}. In the 1P setting, the model was trained with the single key data for the maximum of 1,300 epochs.

As an ablation study, two baseline models were compared with CMT. First model is a vanilla Transformer. It consists of only CE and PD, both of which are stacks of self-attention blocks. Second model replaces the self-attention CE of the first baseline model with a BLSTM. For both baseline models, there are no separate rhythm decoders, and PDs are responsible for predicting pitch tokens directly.

V. EVALUATION

A. QUANTITATIVE EVALUATION

To assess whether or not melody generation models have been trained well, generated melodies were quantitatively evaluated in the following four ways: examining token accuracy and loss value, analyzing chord tone ratio, using MGEval framework [37], and investigating bar rhythms.

1) VALIDATION ACCURACY AND LOSS

By referring to [38], [39], accuracy of predicting a token at the next time step was employed as an evaluation metric for model selection. For the two-phase training of CMT, RD with the highest validation rhythm accuracy, which mostly corresponds to the model with the smallest loss value, was chosen as the pre-trained model during the second phase.

While low pitch accuracy doesn't necessarily mean that the generated melody is not in harmony, high validation pitch accuracy can be interpreted as the model's ability to generate melody that is harmonious with the chord progression. Two types of pitch accuracies were calculated. The first type considers all the 50 pitch tokens, including hold and rest. This is an adequate measure for comparing the two baseline models. However, predicting hold and rest tokens at the pitch decoding step is trivial for CMT since PD takes the ground truth rhythm sequence as input. When comparing different CMT settings, the accuracy obtained without those two tokens

TABLE 1. Validation accuracies and loss values of two baseline models and CMT in different experiment settings. The total loss $L = L_r + L_p$ is the sum of rhythm loss L_r and pitch loss L_p.

Model	Setting	Accuracy(%)			Loss		
		Rhythm	Pitch: Including hold & rest	Pitch: Onset only	Rhythm loss L_r	Pitch loss L_p	Total loss L
Baseline 1 (Self-attention CE + PD)		-	78.2	14.3	-	0.511	
Baseline 2 (BLSTM CE + PD)		-	79.8	22.5	-	0.459	
CMT (BLSTM CE + RD + PD)	1P	86.6	86.5	35.1	0.317	0.315	0.632
	2P-SK-RO	86.2	85.0	28.9	0.326	0.405	0.731
	2P-PV-RO	89.3	86.1	38.6	0.262	0.285	0.547
	2P-SK-RP	86.3	86.2	32.9	0.322	0.338	0.660
	2P-PV-RP	**89.6**	**87.7**	**47.2**	**0.256**	**0.238**	**0.494**

reveals the performance of PD more clearly. Accordingly, to compute this second type of accuracy over the 48 onset pitch tokens, the number of correctly predicted pitch tokens was counted only when the corresponding rhythm token was an onset.

The results of two baseline models and five settings for CMT settings are reported in Table 1. Higher pitch accuracies and lower pitch loss values of baseline 2 over baseline 1 explain the reason for choosing BLSTM as an encoder for CMT, instead of self-attention blocks. All the CMT settings yielded better results with respect to pitch, suggesting the effectiveness of a separate rhythm decoder.

Comparisons between PV and SK settings show that training with the pitch varied rhythm data at the first phase resulted in improvements in terms of both rhythm and pitch. 2P-PV-RO and 2P-PV-RP achieved higher accuracies than their counterparts, 2P-SK-RO and 2P-SK-RP, respectively. Even though the settings differ in training of the first phase and only the RD's parameters were retained at the second phase, their pitch accuracies exhibit notable differences. Accordingly, it can be inferred that accurate RD helps the training of PD. The fact that the results of RP settings were better than those of RO settings also supports the claim. It can be interpreted as the effect of sharing intermediate rhythm representations from RD at pitch decoding step.

2) CHORD TONE RATIO

To assess how well the generated melodies comply with a given chord progression, we compared the chord tone ratios for the test dataset and those for the results by CMT, EC^2-VAE [29], and the two baseline models. While the goal of Yang *et al.* [29] was not melody generation, EC^2-VAE was considered as an alternative since it can produce melodies from scratch by sampling latent variables from a latent space, and in particular it took an approach most similar to ours in terms of chord conditioning and separation of rhythm and pitch.

CMT, EC^2-VAE, and the two baseline models were all trained with EWLD to generate melodies of 8 bars. Conditioned on the chord progressions of the 2,950 test data instances, 2,950 samples were constructed from each model. Under the intuition that a harmonic melody would have chordal notes especially on the first strong beat, we also computed the chord tone ratio of the first beat of each bar, along with the overall ratio.

TABLE 2. Mean values of chord tone ratios.

Model	Chord tone ratio(%): Overall	Chord tone ratio(%): 1st beat
Dataset	71.42	79.61
Baseline 1 (Self-attention CE + PD)	50.11	45.37
Baseline 2 (BLSTM CE + PD)	68.88	65.78
CMT (BLSTM CE + RD + PD)	72.53	80.63
EC^2-VAE	63.19	65.99

The average chord tone ratios computed from the test dataset, CMT with 2P-PV-RP setting, two baseline models and EC^2-VAE are summarized in Table 2. For the test data of EWLD dataset, 71.42% of notes were one of the notes that make up the chord on their onset time on average. The average chord tone ratios of two baseline models and EC^2-VAE were 50.43%, 68.82%, and 63.19%, respectively, which are clearly lower than that of the test dataset. Conversely, the average ratio of CMT was 72.53%, similar to that of the dataset. This tendency appears to be more evident in the ratios for the first beats. From this result, it can be concluded that CMT was trained to make good use of the chord information and produce melody that is harmonious with the chord to a similar level to that of the dataset.

3) MGEval FRAMEWORK

MGEval (Music Generation Evaluation) is a framework developed by Yang and Lerch [37] to evaluate generated music. It extracts nine metrics from a piece of music, five of which are pitch-based features and four related to rhythm. Each feature name and its abbreviation are specified in Table 3.

TABLE 3. Features extracted in MGEval framework [37].

Feature Type	Feature Name
Pitch based features	Pitch count (PC)
	Pitch class histogram (PCH)
	Pitch class transition matrix (PCTM)
	Pitch range (PR)
	Pitch interval (PI)
Rhythm based features	Note count (NC)
	Average inter onset interval (IOI)
	Note length histogram (NLH)
	Note length transition matrix (NLTM)

TABLE 4. Results of applying the MGEval framework to the test dataset and the outputs by CMT and EC^2-VAE, respectively. For $\mathcal{M} \in \{\mathcal{M}_{CMT}, \mathcal{M}_{EC2}\}$ and feature f, KLD indicates the Kullback-Leibler divergence between the two PDFs estimated from $\delta^f(\mathcal{M}_{test}, \mathcal{M})$ and $\delta^f(\mathcal{M}_{test}, \mathcal{M}_{test})$, and OA stands for the overlapping area.

	Dataset		CMT				EC^2-VAE			
	Mean	STD	Mean	STD	$\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{CMT}})$ KLD	OA	Mean	STD	$\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{EC}^2})$ KLD	OA
PC	8.26	2.26	7.51	1.76	$\mathbf{9.75\times10^{-3}}$	**0.881**	6.43	1.91	1.45×10^{-2}	0.876
PCH	-	-	-	-	$\mathbf{1.87\times10^{-3}}$	**0.983**	-	-	5.42×10^{-1}	0.674
PCTM	-	-	-	-	$\mathbf{8.95\times10^{-3}}$	**0.968**	-	-	1.07×10^{-2}	0.943
PR	12.26	3.21	11.51	2.80	$\mathbf{5.64\times10^{-3}}$	**0.900**	11.60	4.46	9.43×10^{-2}	0.817
PI	2.36	0.71	2.16	0.55	$\mathbf{7.16\times10^{-3}}$	**0.945**	2.94	1.35	2.98×10^{-2}	0.829
NC	28.49	9.21	27.49	9.47	$\mathbf{7.42\times10^{-3}}$	**0.973**	29.53	13.28	5.80×10^{-2}	0.855
IOI	0.28	0.10	0.28	0.10	$\mathbf{3.95\times10^{-2}}$	**0.979**	0.29	0.12	6.09×10^{-2}	0.970
NLH	-	-	-	-	$\mathbf{9.87\times10^{-4}}$	**0.984**	-	-	2.02×10^{-2}	0.918
NLTM	-	-	-	-	$\mathbf{1.78\times10^{-2}}$	**0.984**	-	-	8.94×10^{-2}	0.871

Based on MGEval, we measured how much the statistics of the generated music agree with those of the dataset through carrying out pairwise cross-validation. We applied the evaluation framework to the EWLD's test dataset as well as the music pieces generated by models, described in Section V-A2. For all the features considered, the Euclidean distances between all the possible pairs of two music samples were computed.

Let $\mathcal{M}_{\text{test}}$, $\mathcal{M}_{\text{CMT}}$, and $\mathcal{M}_{\text{EC}^2}$ respectively denote the test dataset and the sets of music pieces produced by CMT and EC^2-VAE, respectively. d_{ij}^f denotes the Euclidean distance between the pieces i and j with respect to feature f. For feature f, two distance sets $\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{CMT}}) = \{d_{ij}^f | i \in \mathcal{M}_{\text{test}}, j \in \mathcal{M}_{\text{CMT}}, \ \forall i, j\}$ and $\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{test}}) = \{d_{ij}^f | i \in \mathcal{M}_{\text{test}}, j \in \mathcal{M}_{\text{test}}, \ \forall i, j \text{ s.t. } i \neq j\}$ were converted into probability density functions (PDFs), to obtain Kullback-Leibler divergence (KLD) [40] as well as overlapping area (OA) values between the two PDFs. If CMT was trained to be able to capture the characteristics of the dataset well in terms of feature f, $\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{CMT}})$ and $\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{test}})$ would follow similar distributions, resulting in small KLD and large OA. KLD and OA between $\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{EC}^2})$ and $\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{test}})$ were computed in the same way.

Table 4 shows means and standard deviations for the features (except for histograms and matrices) of the test dataset, CMT, and EC^2-VAE, along with the results of distribution comparison between the test dataset and each of the two models. The results show that CMT outperformed EC^2-VAE in terms of both KLD and OA for all the features considered. Visualization results for the PDFs converted from three distance sets, $\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{test}})$, $\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{CMT}})$, and $\delta^f(\mathcal{M}_{\text{test}}, \mathcal{M}_{\text{EC}^2})$, can be found in the supplementary material.

4) BAR RHYTHM ANALYSIS

Furthermore, distributions of bar rhythms were examined. In our data representation, one bar consists of sixteen time steps and there are three kinds of rhythm tokens. Therefore, the number of possible rhythm patterns in a bar is 3^{16}. For every instance of 8 bars in the test dataset and a piece of music produced by CMT and EC^2-VAE, we counted the number of different bar rhythms and compared the Jensen-Shannon divergence [41] between distributions. Only CMT and EC^2-VAE were considered and not the baseline models mentioned above, since the baseline models do not predict rhythm tokens explicitly.

There were 1,219 different bar rhythms in the test dataset. For the generated samples by CMT and EC^2-VAE, the numbers of different bar rhythms were 2,259 and 3,532, respectively. The total number of different bar rhythms in all three sets was 5,419.

The Jensen-Shannon divergence [41] between the test dataset and outputs by CMT was 8.14×10^{-2}, which was smaller than 2.17×10^{-1}, the divergence between the test dataset and EC^2-VAE's outputs. This indicates that the distribution of bar rhythms generated by CMT is more similar to that of the dataset.

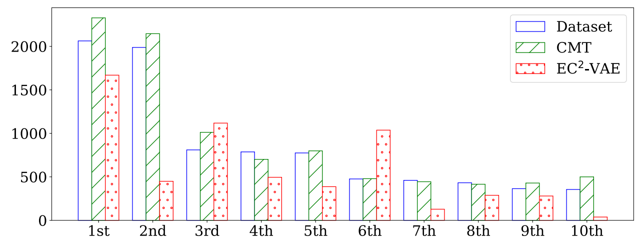


FIGURE 3. Bar chart of rhythm patterns in the test dataset, CMT, and EC^2-VAE. 10 most frequent rhythms from the test data are sorted on the *x*-axis by their frequencies and the *y*-axis indicates the counts of each bar rhythm.

Figure 3 depicts the results in more detail. Out of 5,419 patterns, 10 most frequent bar rhythms in the test dataset are illustrated in bar charts. The dataset and CMT show a similar tendency while EC^2-VAE does not. Specifically, for the 2nd most frequent bar rhythm, the pattern occurred for almost 2,000 times in the dataset and CMT's generations, whereas the occurrence was less than 500 for EC^2-VAE. On the other hand, for the 6th most frequent bar rhythm, the frequency was less than 500 in the dataset and CMT's generation results but more than 1,000 in generation results by EC^2-VAE.

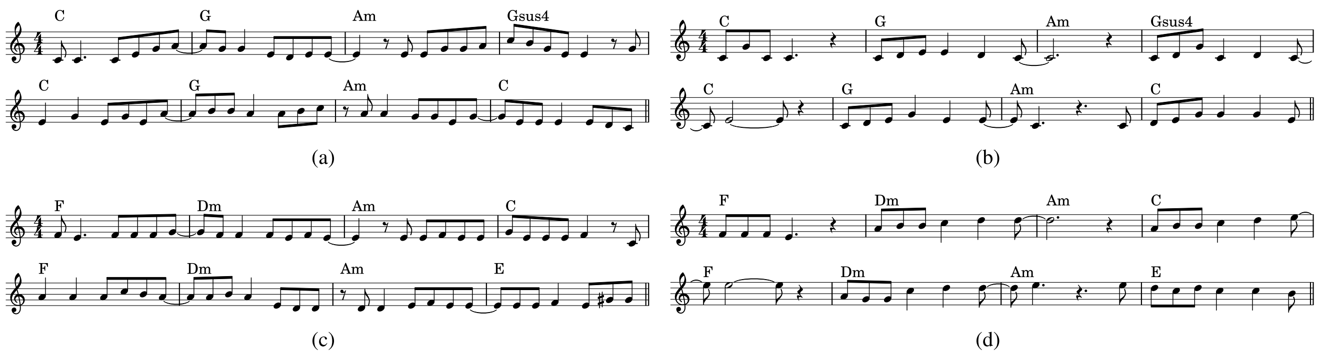



FIGURE 4. **Melodies generated in accordance with chords in major and minor keys. (a) and (b) are different melodies generated from a single chord progression in a major key. (c) and (d) are generated for another chord progression in a minor key, conditioned on the rhythms of (a) and (b), respectively.**

B. QUALITATIVE EVALUATION

1) VARIETY AND CHORD ACCORDANCE

Figure 4 demonstrates the capability of CMT to generate various melodies that are consonant with given chord progressions. From the test instances of the custom score dataset described in Section IV-A1, two chord progressions were extracted, one in a major key and the other in a minor key.

Figures 4 (a) and (b) show the scores of two melodies generated by CMT for a single chord progression in a major key. CMT was able to produce different outputs even with the same input due to sampling strategy that substitutes argmax, as explained in Section IV-B. The results of feeding the rhythms of (a) and (b) into PD with a chord progression in a minor key are presented in Figures 4 (c) and (d), respectively. It can be observed that pitch progressions are adjusted appropriately.

2) REPETITIONS AND VARIATIONS

Figure 5 depicts the samples of 8-bar music produced by CMT, conditioned on randomly sampled chord progressions from the custom score test set. Repeated rhythms within each sample are depicted as the boxes with solid lines, and their variations as the boxes of the same color but with dashed lines.

In Figure 5 (a), the rhythm of the dotted 8th - 16th - 8th - 8th notes is repeated, marked as the solid red box. Despite the same rhythm, the notes' pitches differ slightly from each other, leading to avoidance of boredom caused by simple repetition of the same melody. Rhythm itself varies from the original pattern, bordered by the dashed red boxes: consolidation of the second (16th) and the third (8th) notes into the dotted 8th note in the 4th bar, and fragmentation of the first dotted 8th note into the 16th and the 8th notes in the 7th bar. Similar variations can also be observed in the green boxes.

Repetitions that are longer than those of Figure 5 (a) can be found in the orange boxes of Figure 5 (b). The 1st and 5th bars have the same rhythms. The 2nd and 6th bars differ from them only at the 4th beat, changing the 8th note to the rest of the same length. Purple boxes reveal that rhythmic repetition

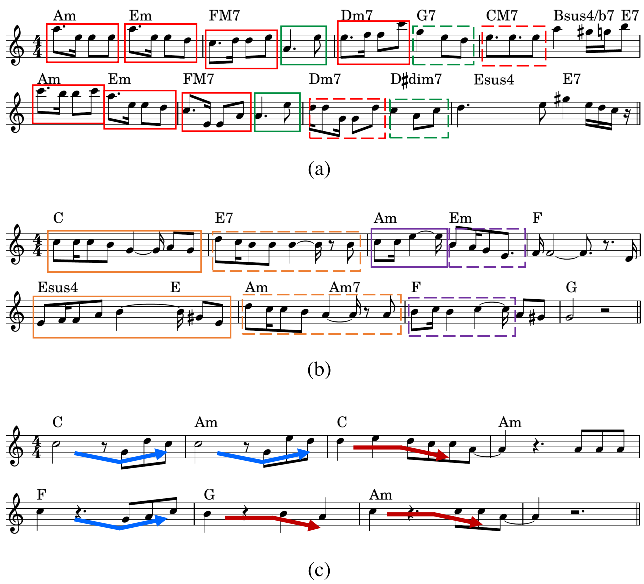


FIGURE 5. **Examples of 8-bar music generated by CMT. Boxes with solid lines represent repeated rhythms and those with dashed lines of the same color indicate their variations. Arrows emphasize similar pitch contours.**

is not just limited to that of a fixed length. Figure 5 (c) shows an example for the repetitions of pitch contours. Two types of pitch contour patterns are repeated in three bars, depicted as the blue and brown arrows, respectively. Note that similar rhythm patterns can be observed in all the bars except the 8th.

3) HUMAN EVALUATION

We trained EC^2-VAE with the custom score dataset as an alternative model to compare with CMT. CMT and EC^2-VAE generated 15 pieces of 8-bar music each, which were conditioned on the chord progressions from randomly sampled test data instances. With additional 15 instances from the test dataset, total 45 pieces of music were the candidates of listening samples. 40 participants listened to three randomly sampled pieces of music from each of the following sets: test dataset, results generated by CMT and EC^2-VAE, respectively. Each participant was asked to rate each sample in a 5-point Likert scale, from 1 (very low) to 5 (very high). Four criteria were chosen by referring to [25], [27], [29]:

1) Rhythm: Whether the rhythm sounds good or not.
2) Harmony: How harmonious the melody and chord progression are.
3) Creativity: How novel the melody is.
4) Naturalness: How much the music sounds like those composed by human.

The results of human evaluation are visualized in Figure 6 as violin plots. CMT's rhythm, pitch, and naturalness scores were higher than those of EC^2-VAE, suggesting that its results sound better in terms of rhythm and pitch, and are also more natural and sound like human made music. As for creativity, CMT got higher scores than real melodies in the dataset. This implies that CMT was not trained to simply copy the training data, and also that the produced melodies are novel.

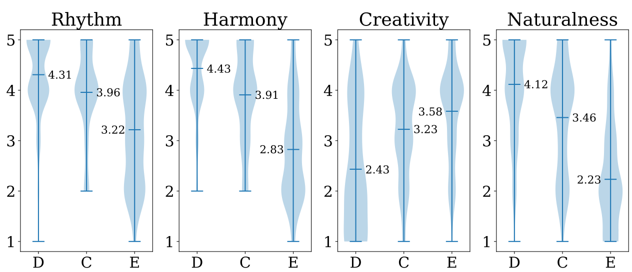


FIGURE 6. Result of human evaluation comparing ground truth dataset (D), CMT (C) and EC^2-VAE (E). The middle bars indicate the mean values.

In conclusion, CMT appears to be able to produce novel melodies that are more natural and sound like human made than those generated by EC^2-VAE.

VI. CONCLUSION

This paper introduced a novel chord conditioned K-POP melody generation model, named CMT, and proposed training methods to improve its performance. The model generates rhythm first, and then pitch sequence corresponding to the rhythm. Comparisons between different experiment settings proved that dividing the training procedure into two phases and training with the pitch varied rhythm data were effective in improving the accuracy of both rhythm and pitch. Evaluations with several quantitative metrics implied that the distributions of the test dataset overlap more with those of the proposed model's results than those of the alternatives considered in terms of both pitch and rhythm based features, including chord tone ratios and bar rhythms. Examples of generated music demonstrated the model's ability to adaptively generate various melodies with repetitive and varying patterns for a given chord. The results of human listening test demonstrated that the melodies generated by the proposed model were novel, harmonious, natural in rhythm, and sound like music composed by human.

Yet, one of the drawbacks of the proposed work is that the length of generated music is limited to 8 bars. For a generative model to produce real K-POP melodies, it should be capable of generating melodies with repetitions and variations over longer time spans. Song generation with consideration of structures such as verse and chorus is also necessary. CMT itself can be trained with longer songs, or can be utilized in other scenarios such as constructing a chorus from a verse. We leave the task of generating longer melodies for future work.

REFERENCES

[1] K. Markov and T. Matsui, "Music genre and emotion recognition using Gaussian processes," *IEEE Access*, vol. 2, pp. 688–697, 2014, doi: 10.1109/ACCESS.2014.2333095.

[2] P.-H. Kuo, T.-H.-S. Li, Y.-F. Ho, and C.-J. Lin, "Development of an automatic emotional music accompaniment system by fuzzy logic and adaptive partition evolutionary genetic algorithm," *IEEE Access*, vol. 3, pp. 815–824, 2015, doi: 10.1109/ACCESS.2015.2443985.

[3] I. Goienetxea, I. Mendialdua, I. Rodriguez, and B. Sierra, "Statistics-based music generation approach considering both rhythm and melody coherence," *IEEE Access*, vol. 7, pp. 183365–183382, 2019, doi: 10.1109/ACCESS.2019.2959696.

[4] R. Yang, L. Feng, H. Wang, J. Yao, and S. Luo, "Parallel recurrent convolutional neural networks-based music genre classification method for mobile devices," *IEEE Access*, vol. 8, pp. 19629–19637, 2020, doi: 10.1109/ACCESS.2020.2968170.

[5] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," 2018, *arXiv:1810.12247*. [Online]. Available: http://arxiv.org/abs/1810.12247

[6] R. Sabathe, E. Coutinho, and B. Schuller, "Deep recurrent music writer: Memory-enhanced variational autoencoder-based musical score composition and an objective measure," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 3467–3474.

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

[8] 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*, New Orleans, LA, USA, 2018, pp. 34–41.

[9] 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. ICML*, Stockholm, Sweden, 2018, pp. 4364–4373.

[10] I. Simon, A. Roberts, C. Raffel, J. Engel, C. Hawthorne, and D. Eck, "Learning a latent space of multitrack measures," 2018, *arXiv:1806.00195*. [Online]. Available: http://arxiv.org/abs/1806.00195

[11] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer," in *Proc. ISMIR*, Paris, France, 2018, pp. 747–754.

[12] N. Jaques, S. Gu, D. Bahdanau, J. M. Hernández-Lobato, R. E. Turner, and D. Eck, "Sequence tutor: Conservative fine-tuning of sequence generation models with KL-control," 2016, *arXiv:1611.02796*. [Online]. Available: http://arxiv.org/abs/1611.02796

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, Long Beach, CA, USA, 2017, pp. 6000–6010.

[14] Q. Guo, J. Huang, N. Xiong, and P. Wang, "MS-pointer network: Abstractive text summary based on multi-head self-attention," *IEEE Access*, vol. 7, pp. 138603–138613, 2019, doi: 10.1109/ACCESS.2019.2941964.

[15] S. Shang, J. Liu, and Y. Yang, "Multi-layer transformer aggregation encoder for answer generation," *IEEE Access*, vol. 8, pp. 90410–90419, 2020, doi: 10.1109/ACCESS.2020.2993875.

[16] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," 2018, *arXiv:1809.04281*. [Online]. Available: http://arxiv.org/abs/1809.04281

[17] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, "A Bi-directional transformer for musical chord recognition," in *Proc. ISMIR*, Delft, The Netherlands, 2019, pp. 620–627.

[18] Q. Lin, Y. Niu, Y. Zhu, H. Lu, K. Z. Mushonga, and Z. Niu, "Heterogeneous knowledge-based attentive neural networks for short-term music recommendations," *IEEE Access*, vol. 6, pp. 58990–59000, 2018, doi: 10.1109/ACCESS.2018.2874959.

[19] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using BLSTM networks," in *Proc. ISMIR*, Suzhou, China, 2017, pp. 621–627.

[20] H. Chu, R. Urtasun, and S. Fidler, "Song from PI: A musically plausible network for pop music generation," 2016, *arXiv:1611.03477*. [Online]. Available: http://arxiv.org/abs/1611.03477

[21] W. Yang, P. Sun, Y. Zhang, and Y. Zhang, "CLSTMS: A combination of two LSTM models to generate chords accompaniment for symbolic melody," in *Proc. Int. Conf. High Perform. Big Data Intell. Syst. (HPBD&IS)*, May 2019, pp. 176–180.

[22] B.-S. Lin and T.-C. Yeh, "Automatic chord arrangement with key detection for monophonic music," in *Proc. Int. Conf. Soft Comput., Intell. Syst. Inf. Technol. (ICSIIT)*, Sep. 2017, pp. 21–25.

[23] G. Brunner, Y. Wang, R. Wattenhofer, and J. Wiesendanger, "JamBot: Music theory aware chord based generation of polyphonic music with LSTMs," in *Proc. IEEE 29th Int. Conf. Tools With Artif. Intell. (ICTAI)*, Nov. 2017, pp. 519–526.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.

[25] L. C. Yang, S. Y. Chou, and Y. H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proc. ISMIR*, Suzhou, China, 2017, pp. 324–331.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[27] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, "XiaoIce band: A melody and arrangement generation framework for pop music," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2837–2846.

[28] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–Decoder approaches," in *Proc. SSST-8, 8th Workshop Syntax, Semantics Struct. Stat. Transl.*, 2014, pp. 103–111.

[29] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. Xia, "Deep music analogy via latent representation disentanglement," in *Proc. ISMIR*, Delft, The Netherlands, 2019, pp. 596–603.

[30] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., (Short Papers)*, vol. 2, 2018, pp. 464–468.

[31] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2005, pp. 2047–2052, doi: 10.1109/IJCNN.2005.1556215.

[32] T. P. Chen and L. Su, "Harmony transformer: Incorporating chord segmentation into harmony recognition," in *Proc. ISMIR*, Delft, The Netherlands, 2019, pp. 259–267.

[33] B. Bi, C. Li, C. Wu, M. Yan, W. Wang, S. Huang, F. Huang, and L. Si, "PALM: Pre-training an Autoencoding & Autoregressive language model for context-conditioned generation," 2020, *arXiv:2004.07159*. [Online]. Available: http://arxiv.org/abs/2004.07159

[34] F. Simonetta, F. Carnovalini, N. Orio, and A. Rodà, "Symbolic music similarity through a graph-based representation," in *Proc. Audio Mostly Sound Immersion Emotion*, Sep. 2018, p. 26.

[35] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[37] L.-C. Yang and A. Lerch, "On the evaluation of generative models in music," *Neural Comput. Appl.*, vol. 32, no. 9, pp. 4773–4784, May 2020, doi: 10.1007/s00521-018-3849-7.

[38] J. Wu, C. Hu, Y. Wang, X. Hu, and J. Zhu, "A hierarchical recurrent neural network for symbolic melody generation," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2749–2757, Jun. 2020, doi: 10.1109/TCYB.2019.2953194.

[39] F. Colombo, A. Seeholzer, and W. Gerstner, "Deep artificial composer: A creative neural network model for automated melody generation," in *Proc. EvoMUSART*, Amsterdam, The Netherlands, 2017, pp. 81–96.

[40] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[41] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, Jan. 1991, doi: 10.1109/18.61115.

KYOYUN CHOI received the B.S. degree in industrial engineering from Seoul National University (SNU), South Korea, in 2016, where he is currently pursuing the Ph.D. degree with the Information Management Laboratory, Department of Industrial Engineering. His current research interests include generative artificial intelligence, deep learning applications, and symbolic music generation.

JONGGWON PARK received the B.S. degree in industrial engineering from Seoul National University (SNU), South Korea, in 2018, where he is currently pursuing the Ph.D. degree with the Information Management Laboratory, Department of Industrial Engineering. His current research interests include audio signal processing, sequence to sequence learning, and deep learning applications.

WAN HEO received the B.S. degree in industrial engineering from Seoul National University (SNU), South Korea, in 2017, where he is currently pursuing the Ph.D. degree with the Information Management Laboratory, Department of Industrial Engineering. His current research interests include sequence to sequence learning, deep learning generation, and audio signal processing.

SUNGWOOK JEON received the B.S. degree in industrial engineering from Seoul National University (SNU), South Korea, in 2012, where he is currently pursuing the Ph.D. degree with the Information Management Laboratory, Department of Industrial Engineering. His current research interests include sequence generation models and deep learning applications.

JONGHUN PARK received the Ph.D. degree in industrial and systems engineering from the Georgia Institute of Technology, Atlanta, in 2000, with a focus on computer science. He is currently a Professor with the Department of Industrial Engineering, Seoul National University (SNU), South Korea. Before joining SNU, he was an Assistant Professor with the School of Information Sciences and Technology, Pennsylvania State University, University Park, and the Department of Industrial Engineering, KAIST, Daejeon. His research interests include generative artificial intelligence and deep learning applications.