POP MUSIC GENERATION WITH CONTROLLABLE PHRASE LENGTHS

Daiki Naruse[1] **Tomoyuki Takahata**[1] **Yusuke Mukuta**[1,2] **Tatsuya Harada**[1,2]
[1] The University of Tokyo, Japan
[2] RIKEN, Japan
{naruse, takahata, mukuta, harada}@mi.t.u-tokyo.ac.jp

ABSTRACT

Research on music generation using deep learning has attracted more attention; in particular, Transformer-based models have succeeded in generating coherent musical pieces. Recently, an increasing number of studies have focused on phrases that are smaller musical units, and several studies have addressed phrase-level control. In this study, we propose a method for sequentially generating a piece that enables the control of each phrase length and, consequently, the length of the entire piece. We added PHRASE and a new event, BAR COUNTDOWN, which indicates the number of bars remaining in the phrase, to the existing event-based music representations. To reflect user input indicating the phrase lengths of the piece being generated, we used an autoregressive generation model that adds these two events to the generated event-token sequence based on the user input and uses it as input for the next time step. Subjective listening tests revealed that the pieces generated by our methods possessed designated phrase lengths and ended naturally at the determined length. [1]

1. INTRODUCTION

Music generation has been studied for more than half a century [1] and has advanced significantly in recent years with the development of deep learning. In many studies, deep neural sequence models such as recurrent neural networks (RNNs) and Transformers [2] have been used to model music. Transformer-based methods [3–7] have succeeded in generating coherent music throughout a piece. To apply these sequence models to music generation, it is necessary to represent a piece as a sequence of tokens. Event-based representations such as MIDI-like [8] and its advanced versions, REMI [6] and CP [7], have been used.

More recently, an increasing number of studies have focused on phrases and sections [9–14], which are smaller musical segments. These studies aimed to generate a structured piece that was divided into several segments and developed through repetition and transformation. Several studies addressed phrase-level control [11–13] and allowed the control of phrase attributes such as melody, rhythm, and harmonic fullness. Length is another important phrase attribute and is controllable with a phrase-by-phrase generation policy [13], which means that each phrase is generated independently and joined. However, this phrase-by-phrase generation policy has a limitation in that natural transitions between phrases are not guaranteed.

Therefore, we worked on controlling the phrase lengths with a sequential generation policy. The sequential generation policy, unlike the phrase-by-phrase or section-by-section generation policy [9, 13], is a method of sequentially generating an entire piece at once. We aim to create a model that outputs a piece according to user input regarding the configuration of the phrases and the length of each phrase, as shown in Figure 1 (the detailed generation process is described in Sections 3.3 and 3.4). The controllability of each phrase length implies that the length of the entire piece can be controlled. To control the length of each phrase and the entire piece, we extended two recently used event-based music representations, REMI [6] and CP [7]. The random timing of the switching of phrases and the end of the generation in the existing representations is likely due to the model not knowing which phrase and where it is generating. Therefore, we added PHRASE and a new event, BAR COUNTDOWN, which indicates the number of bars remaining in the phrase, to REMI and CP. To reflect the user input, we used an autoregressive generation method in which these two events were added based on the user input to the generated event-token sequence, and the sequence was entered into the model again. To evaluate this approach, two subjective listening tests were conducted for the length of each phrase and the entire piece. By comparing our methods with the dataset and existing methods, we demonstrate our methods are effective in length control.

Our contributions are summarized as follows:

- We extended the existing music representations (REMI [6] and CP [7]) by adding PHRASE and BAR COUNTDOWN events and showed that both events are necessary for length control.
- We enabled the reflection of the user input by an autoregressive generative model that adds the PHRASE and BAR COUNTDOWN events to the generated event-token sequence based on the user input and uses it as input for the next time step.

[1] Samples of the generated pieces are available at https://mil-tokyo.github.io/phrase-length-designated-music-generation/.

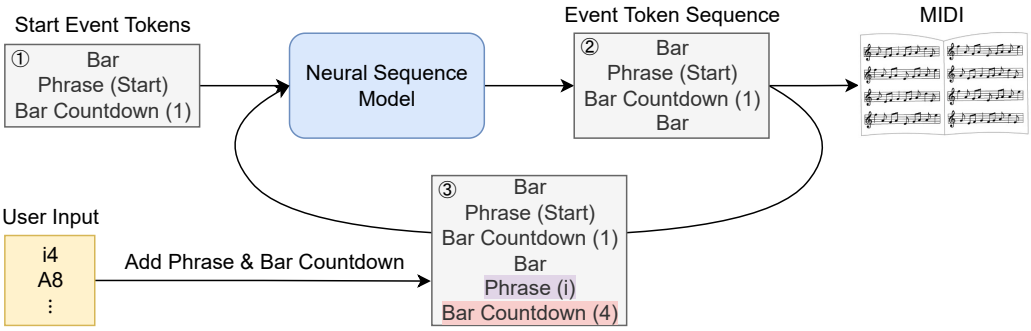


Figure 1: Generation process reflecting user input regarding phrase lengths.

2. RELATED WORK

2.1 Event-based Music Representations

To apply neural sequence models to music generation, music must be represented using a token sequence. Many studies [3–5, 15] adopted MIDI-like [8]. NOTE ON and NOTE OFF events indicate the start and end of a note, respectively, and a TIME SHIFT event advances the time step.

In MIDI-like, bars and beats are implicit, which are clearly indicated in the score, and it is difficult for the model to learn beat regularity and rhythmic structure. The model also has difficulty learning that the NOTE ON and NOTE OFF events must exist in pairs. To address these problems, an improved music representation called REMI (revamped MIDI-derived events) [6] was proposed. In REMI, the TIME SHIFT event in MIDI-like is replaced with BAR and BEAT events, and the NOTE OFF event is replaced with a NOTE DURATION event. TEMPO and CHORD events are added for clear harmony and expressive rhythmic freedom.

Later, a further extension of REMI called CP (compound word representation) [7] was suggested. In CP, consecutive and related events are grouped and placed in the same time step. Specifically, BAR, BEAT, CHORD, and TEMPO events are grouped into a METRICAL family and note-related events into a NOTE family. Additionally, a new event, EOS, is added to mark the end of a piece.

2.2 Latest Transformer-based Music Generation

Recently, Transformer-based methods have successfully generated coherent music throughout a piece and have become common in automatic composition in the symbolic domain. The Music Transformer study [3] was the first to apply the Transformer model to music generation. This study used MIDI-like to generate pieces by autoregressively predicting an event token at each time step. The Pop Music Transformer study [6] proposed REMI and generated pieces with a better rhythmic structure. The Jazz Transformer study [16] addressed the generation of Jazz. An attempt was made to introduce structure by adding the following four structure-related events to REMI: PHRASE, MLU, PART, and REPETITION. The CP Transformer study [7] proposed CP. Predicting events of the same family simultaneously at each time step significantly reduces the length of the token sequence, resulting in faster learning and inference. In addition, the EOS event allowed the model to complete the generation with the natural closure.

HAT (Harmony-Aware Hierarchical Music Transformer) [14] focused on phrases and sections. It represents music in CP and uses three Transformers hierarchically to allow event tokens to interact at different levels. The structure of the pieces was improved by learning the texture and the form jointly bridged by the harmony.

MusicFrameworks [13] is a monophonic melody generation system that enables phrase-level control. Music is described using music frameworks, a hierarchical music structure representation, and melodies are generated through a multi-level generative process. The manipulation of music frameworks allows control over phrase attributes such as structure, melody, and rhythm. Phrase length can also be controlled by modifying the structural information and length of the rhythm and chord information in each phrase. However, because of the phrase-by-phrase generation policy, there is no guarantee that transitions between phrases are natural.

3. METHOD

3.1 Phrase-Length Designated Music Generation

In this study, we worked on an automatic composition task that allowed control over the length of each phrase (and the length of the entire piece) with a sequential generation policy. The user inputs the configuration of the phrases and the length of each phrase in units of bars, and a piece with the designated phrase lengths and total length is generated. For example, if the input is "i4 A8 B8 o4," then a piece is generated with four bars for the intro phrase, eight bars for phrase A, eight bars for phrase B, and four bars for the outro phrase, with a total length of 24 bars.

3.2 PHRASE and BAR COUNTDOWN Events

We added the following two events to REMI [6] and CP [7]: PHRASE and BAR COUNTDOWN.

The first one, PHRASE, is an event that indicates to which phrase a bar belongs, e.g., PHRASE (i) refers to the intro phrase. This event was first proposed in the Jazz Transformer study [16]. In our study, PHRASE (Start) and PHRASE (End) were used in addition to phrase labels in the

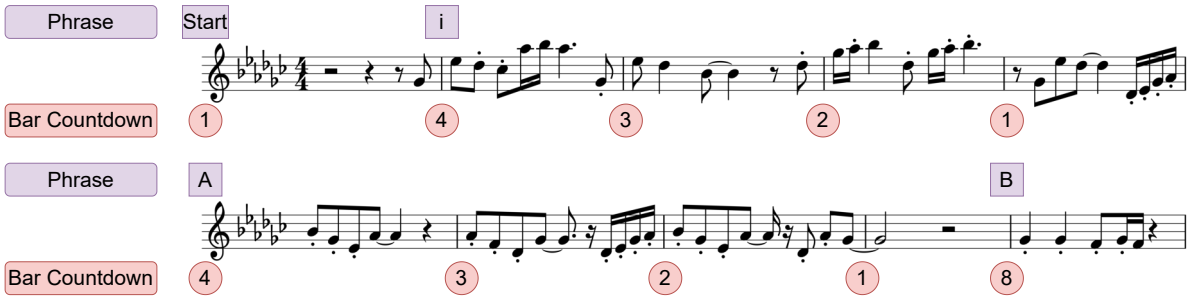


Figure 2: PHRASE and BAR COUNTDOWN events in the staff notation. This is the preprocessed melody part of '001.wav' in POP909 [17].

score. PHRASE (Start) represents one bar before the piece begins. This bar exists in all pieces to consider an anacrusis (or auftakt). Pieces that begin with an anacrusis have some notes in this bar, while those without an anacrusis promptly move to the next bar. PHRASE (End) is placed at the end of the event-token sequence and represents the end of the piece.

The second event, BAR COUNTDOWN, indicates the number of bars remaining in a phrase. If four bars remain, it is expressed as BAR COUNTDOWN (4), and the number of bars is counted for each bar.

These two events are expected to allow the model to know which phrase and where it is generating and to adjust the generation toward the turn of the phrase and the end of the piece. The correspondence between the two events and the musical score is shown in Figure 2. In REMI, the PHRASE and BAR COUNTDOWN events are placed just after the BAR event, which represents a bar line. In CP, these two events are placed along all the time steps. We slightly modified the original settings of CP in the CP Transformer [7] to decompose the METRICAL family into BAR and POS families. The BAR family represents a bar line instead of the BAR event, and the POS family groups the BEAT and CHORD events. In both REMI and CP, performance-related events, NOTE VELOCITY and TEMPO, were not used to reduce the burden of learning. We refer to these extended REMI and CP as **REMI + Ph&BC** and **CP + Ph&BC**, respectively (Ph and BC represent PHRASE and BAR COUNTDOWN, respectively). A list of events used in each representation is shown in Table 1, and an example of each event-token sequence is shown in Figure 3.

3.3 Reflection of User Input

We propose an autoregressive generation method to reflect user input regarding phrase lengths when generating pieces. As shown in Figure 1, event tokens are first predicted using the trained neural sequence model. Before using the predicted event-token sequence as the model input, the PHRASE and BAR COUNTDOWN events are added to the appropriate places based on the user input. This method is expected to enable the input of the phrase lengths that do not exist in the training data because they are present in BAR COUNTDOWN events.

REMI + Ph&BC	**CP + Ph&BC**	
Event	**Event**	**Family**
Note On Note Duration	Note On Note Duration	Note
Bar	-	Bar
Beat Chord	Beat Chord	Pos
Phrase **Bar Countdown**	**Phrase** **Bar Countdown**	[All]
-	Conti	[All]

Table 1: Events in REMI + Ph&BC and CP + Ph&BC.

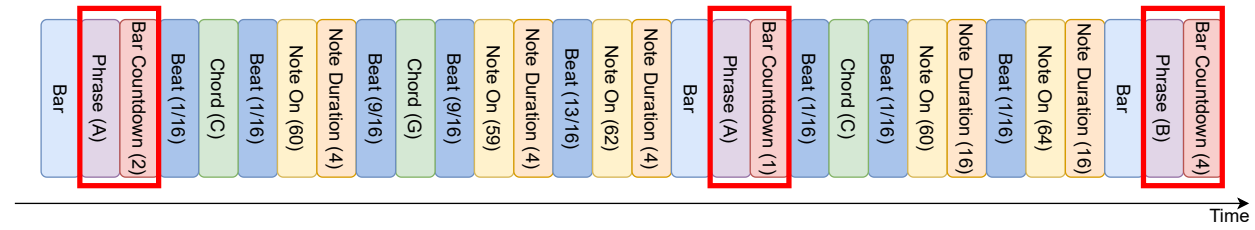
3.4 Pipeline

The same Transformer-based model was used as in the Pop Music Transformer [6] and the CP Transformer [7]. During the training stage, the MIDI file, chord annotation, and phrase annotation of each piece in the dataset are converted into the REMI + Ph&BC or CP + Ph&BC event-token sequence. The model is trained to predict the next event tokens from the input event-token sequence. The pipeline during the generation stage is illustrated in Figure 1. First, BAR, PHRASE (Start), and BAR COUNTDOWN (1) are input to the model to start the generation. Then, as described in Section 3.3, the next event tokens are predicted by the model, and after adding the PHRASE and BAR COUNTDOWN events based on user input, the event-token sequence is again entered into the model. By repeating this process to predict the event tokens sequentially, a piece is generated probabilistically. Once the model has generated the designated number of bars, it adds PHRASE (End) and terminates the generation.

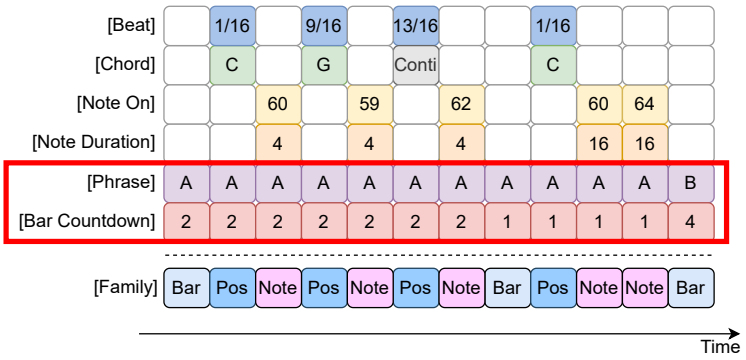
4. EXPERIMENTS

4.1 Dataset

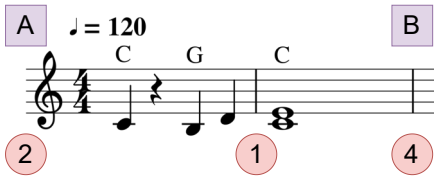
In this study, POP909 [17] is used as the MIDI dataset. The dataset consists of 909 pieces that are piano arrangements of pop songs by professional musicians and is divided into three parts: vocal melody, secondary melody or lead instrument melody, and piano accompaniment. We also used algorithmic chord annotations included in POP909 and the



(a) REMI + Ph&BC



(b) CP + Ph&BC



(c) Musical score

Figure 3: Examples of REMI + Ph&BC (a) and REMI + Ph&BC (b) event-token sequences and the corresponding staff notation (c).

human phrase annotations provided by Dai et al. [18].

After selecting pieces with 4/4 time signatures and excluding those whose downbeats were not aligned with the bar lines, 763 MIDI files were obtained. As a preprocessing step, we merged the three parts and quantized each piece into a 16th-note grid. Next, we shifted all pieces so that the first complete bar was the second to consider the pieces with an anacrusis, as described in Section 3.2. Furthermore, as the number of the intro and outro phrases was smaller than that of the other phrases, we extracted the first and last parts of the pieces and performed data augmentation by transforming their keys in the range of -3 to $+3$. We also modified the chord annotations. All flats on the root note were changed to sharps, and the chord types were limited to the following six types: maj, min, dim, aug, sus4, and sus2.

4.2 Overview of Evaluation Methods

In this study, the length of each phrase and the entire piece was evaluated. In the evaluation of each phrase length, we determined whether each phrase had a designated length by locating the boundary at which the phrase changed. The controllability of the overall length was determined based on whether it ended naturally or abruptly.

For an objective evaluation, methods that can automatically detect phrase boundaries and calculate a naturalness score for the end of a piece are required; however, no suitable methods are available (details are discussed in Section 4.6). In this study, we did not conduct an objective evaluation; rather, we conducted a subjective evaluation.

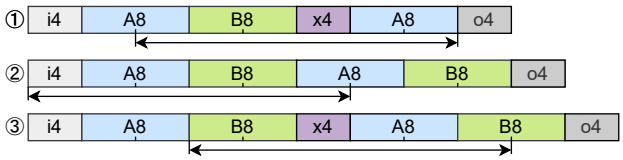
For the subjective evaluation, we administered two listening tests: one to divide a piece into several phrases and the other to evaluate the naturalness of the end. The details are provided in Section 4.4.

4.3 Comparative Methods

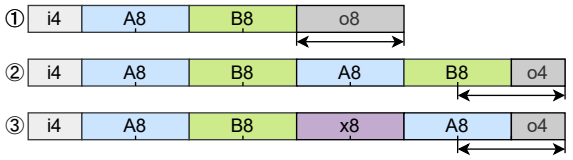
First, we compared our REMI + Ph&BC and CP + Ph&BC with the existing music representations, REMI [6] and CP [7]: **REMI** and **CP**. Since these cannot control the phrase lengths, we evaluated them only for closure. In these methods, we used the events and families shown in Table 1, excluding PHRASE and BAR COUNTDOWN, for comparison under the same conditions. In REMI, generation cannot be terminated by the model; instead, the model is forced to terminate when the number of generated bars reaches the target number. In CP, the model can naturally end a piece, although it cannot control the length of the piece. The final parts of the generated pieces were used for the evaluation.

Next, for the ablation studies, we compared REMI + Ph&BC and CP + Ph&BC with methods with a lower number of events. First, we compared REMI + Ph&BC with **REMI + Ph$_{\text{fewer}}$&BC**, a method that places PHRASE events only at the beginning of phrases. Note that in CP, the number of time steps does not change even if the number of PHRASE events is reduced, so CP + Ph$_{\text{fewer}}$&BC was not evaluated. We also compared our method with methods that used only one of the two events: **REMI + BC**, **REMI + Ph**, **CP + BC**, and **CP + Ph**.

The pieces in the POP909 dataset were also evaluated: **POP909**. Additionally, we intentionally created pieces that ended abruptly by cutting them off in the middle: **POP909$_{\texttt{cut}}$**. This method was used only when evaluating the closure.



(a) Test 1



(b) Test 2

Figure 4: Inputs of the generated pieces used for the evaluation. One division represents four bars. The segments indicated by arrows were extracted and used.

4.4 Subjective Evaluation

We conducted the following two online listening tests using Amazon Mechanical Turk:

Test 1 Phrase Boundary Detection
: We investigated whether each phrase had a designated length or not. The participants listened to a piece while looking at the score and divided it into phrases. They were told in advance how many phrases there were, and they were asked to identify the phrase boundaries.

Test 2 4-Grade Evaluation of Closure
: We examined whether the pieces ended naturally or abruptly. The subjects listened to a piece and answered whether the closure was natural or abrupt on a 4-point Likert scale: "Natural," "Somewhat natural," "Somewhat abrupt," and "Abrupt."

Since the majority of phrases in POP909 are four and eight bars in length [12], three inputs, consisting of four- and eight-bar phrases, were used for generation. The concrete inputs are shown in Figure 4. To reduce the burden on the subjects, in Test 1, 24 bars from the middle of a piece containing four phrases, and in Test 2, eight bars from the end were extracted and used for the evaluation. We generated 50 pieces per input and randomly selected two pieces, i.e., six pieces (three inputs $\times$ two pieces) were evaluated for each method.

First, qualification tests were conducted using the dataset to select those who understood each task and performed well, i.e., conformed to the dataset. In Test 1, 24 subjects correctly identified at least six of the nine phrase boundaries for three POP909 pieces, 13 of whom had more than one year of musical experience. In Test 2, 29 subjects answered "Natural" or "Somewhat natural" for two POP909 pieces and "Abrupt" or "Somewhat abrupt" for two $\text{POP909}_{\text{cut}}$ pieces, 12 of whom had more than one year of experience.

We then tested eight methods (except REMI, CP, and $\text{POP909}_{\text{cut}}$) in Test 1 and all 11 methods in Test 2. Each test was performed 60 times, and one piece per method was evaluated per test. In Test 1, the score was based on the percentage of correct answers, whereas in Test 2, "Natural" was scored as 4 points and "Abrupt" as 1 point.

Method	Test 1	Test 2
POP909	0.778	(3.67)
$\text{POP909}_{\text{cut}}$		1.29
REMI + Ph&BC (ours)	**0.633**	**3.00**
REMI + Ph_{fewer}&BC	0.550	2.34
REMI + BC	0.400	1.92
REMI + Ph	0.356	1.27
CP + Ph&BC (ours)	**0.583**	**3.28**
CP + BC	0.461	2.25
CP + Ph	0.306	1.78
REMI		1.35
CP		(3.17)

Table 2: Average percentage of correct answers for phrase boundaries in Test 1 and the average score of the four-grade evaluation of the closure in Test 2. For both tests, higher scores indicate better performance. The bracketed score in Test 2 indicates that it was evaluated with pieces that were not of the designated length.

4.5 Results

The average percentage of correct answers for phrase boundaries in Test 1 and the average score of the four-grade evaluation of the closure in Test 2 are listed in Table 2. In addition, the one-tailed t-test scores comparing our two methods to the other methods are shown in Table 3.

In Test 1, the scores of REMI + Ph&BC and CP + Ph&BC were much higher than 0.130 ($= 3/23$), the score when the phrase boundaries were answered at random. Compared with POP909, the scores were significantly lower (Table 3). It is suggested that our methods are effective in controlling the phrase lengths. When compared to methods used in the ablation studies, our methods scored significantly higher than methods with one of the two events. This indicates that both PHRASE and BAR COUNTDOWN events are required to control the phrase lengths. No significant differences were found between REMI + Ph&BC and REMI + Ph_{fewer}&BC. Thus, to control the phrase lengths, the number of events can be reduced by placing the PHRASE event only at the beginning of the phrase.

In Test 2, our scores exceeded 2.5, which was in the middle of the score range. They were significantly lower than the POP909 score but significantly higher than the $\text{POP909}_{\text{cut}}$ score (Table 3). Furthermore, they did not differ from the CP score, where the generated pieces ended naturally. Therefore, it can be said that our methods can end a piece naturally at a designated length. When compared to the methods used in the ablation studies, our methods scored significantly higher than all other methods. This indicates that both PHRASE and BAR COUNTDOWN events are required to control the length of the piece. These results

Method	Test 1		Test 2	
POP909	$\mu < {\mu_m}^{**}$	$(p = 0.0022)$	$\mu < {\mu_m}^{***}$	$(p = 3.2 \times 10^{-5})$
$\text{POP909}_{\text{cut}}$			$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 2.0 \times 10^{-17})}$
REMI + Ph_{fewer}&BC	$\mu > \mu_m$	$(p = 0.082)$	$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 7.6 \times 10^{-4})}$
REMI + BC	$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 1.1 \times 10^{-4})}$	$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 2.5 \times 10^{-7})}$
REMI + Ph	$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 7.4 \times 10^{-7})}$	$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 1.1 \times 10^{-17})}$
REMI			$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 6.9 \times 10^{-16})}$
CP			$\mu < \mu_m$	$(p = 0.18)$

(a) Results of the t-test between REMI + Ph&BC and other methods.

Method	Test 1		Test 2	
POP909	$\mu < {\mu_m}^{***}$	$(p = 1.6 \times 10^{-4})$	$\mu < {\mu_m}^{***}$	$(p = 5.8 \times 10^{-4})$
$\text{POP909}_{\text{cut}}$			$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 1.0 \times 10^{-31})}$
CP + BC	$\boldsymbol{\mu > {\mu_m}^{*}}$	$\boldsymbol{(p = 0.024)}$	$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 3.3 \times 10^{-9})}$
CP + Ph	$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 1.6 \times 10^{-6})}$	$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 3.2 \times 10^{-17})}$
REMI			$\boldsymbol{\mu > {\mu_m}^{***}}$	$\boldsymbol{(p = 1.6 \times 10^{-27})}$
CP			$\mu > \mu_m$	$(p = 0.22)$

(b) Results of the t-test between CP + Ph&BC and other methods.

Table 3: One-tailed t-test scores comparing our two methods to the other methods in Tests 1 and 2. μ and μ_m denote the average score of our method and each of the other methods, respectively. ${}^{*}p < .05, {}^{**}p < .01, {}^{***}p < .001$. A p-value less than 0.05 was considered statistically significant.

also highlight the importance of placing the PHRASE event in every bar (REMI + Ph&BC), not just at the beginning of the phrase (REMI + Ph_{fewer}&BC).

4.6 Discussion

Comparing the method that uses only the PHRASE event and the one that uses only the BAR COUNTDOWN event, the BAR-COUNTDOWN-only method scored higher, regardless of the test or representation (Table 2). This means that the BAR COUNTDOWN event was more effective in controlling the length of each phrase and the entire piece. This is consistent with the role of the BAR COUNTDOWN event in teaching the model the number of bars until the end of the phrase. The reasons why adding the PHRASE event to BAR-COUNTDOWN-only method would improve scores are inferred as follows. In Test 1, the PHRASE event indicates that the phrase has changed and may play a role in making the boundaries of the phrase more distinct. In Test 2, the event can tell the model that the phrase being generated is the outro phrase, and the piece is almost over.

A two-tailed t-test was performed to determine whether there was a significant difference between REMI + Ph&BC and CP + Ph&BC. The results showed that there was no significant difference between these methods, with $p = 0.42$ for Test 1 and $p = 0.11$ for Test 2. We can say that both representations achieve equally good results. These two events could potentially be used for new event-based representations derived from REMI and CP. In addition, although the Transformer was used as the model in this study, it is expected to be widely applied to sequence models that perform autoregressive generation.

As mentioned in Section 4.2, there are no objective evaluation metrics suitable for this study; therefore, we did not conduct an objective evaluation but only a careful subjective evaluation. Although the fitness scape plot [19, 20] has been used in studies focusing on the generation of music structures [14, 16], it cannot be used for phrases that do not necessarily repeat, as in this study. Although several algorithms have been proposed to determine phrase boundaries [21], they cannot be adopted because of the low correctness rate when applied to POP909 pieces. The development of phrase-segmentation research and the establishment of objective evaluation methods are required.

One limitation of this study is that it was not possible to create repetitions by designating the same phrase labels in the input. For example, if the input is "A4 B4 A4," the two phrases A are completely different. A possible reason is that the model cannot refer to the next phrase label; therefore, the piece is not connected to the beginning of the next phrase, which was previously defined. A mechanism for making such distant phrases with the same label alike is necessary and is an issue for the future.

In addition, the following points need to be addressed in future studies: (1) combining our methods with music theory to achieve more natural pieces, especially in terms of phrase transition and closure. (2) evaluating length diversity because only a few types of lengths were used because of the convenience of the evaluation. (3) controlling other phrase attributes such as emotions.

5. CONCLUSION

In this study, we proposed a method to control the length of each phrase and the entire piece using a sequential generation policy. In this method, two events are added to the existing event-based music representations: the PHRASE event, which indicates the phrase to which the bar belongs, and the BAR COUNTDOWN event, which indicates the number of bars remaining in the phrase. To reflect the user input, an autoregressive generative model is used that adds these two events based on the user input to the previously generated event-token sequence and uses it as input for the next time step. Subjective listening tests indicated that adding two events effectively controlled the length of each phrase and the entire piece.

6. ACKNOWLEDGEMENT

This work was partially supported by JST AIP Acceleration Research JPMJCR20U3, Moonshot R&D Grant Number JPMJPS2011, CREST Grant Number JPMJCR2015, JSPS KAKENHI Grant Number JP19H01115, and Basic Research Grant (Super AI) of Institute for AI and Beyond of the University of Tokyo.

7. 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] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *NeurIPS*, 2017.

[3] 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 *ICLR*, 2019.

[4] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. McAuley, "LakhNES: Improving multi-instrumental music generation with cross-domain pre-training," in *ISMIR*, 2019.

[5] C. Payne, "MuseNet," *OpenAI Blog*, 2019.

[6] Y.-S. Huang and Y.-H. Yang, "Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions," in *ACM Multimedia*, 2020.

[7] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs," in *AAAI*, 2021.

[8] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: learning expressive musical performance," *Neural Computing and Applications*, 2018.

[9] Y. Zhou, W. Chu, S. Young, and X. Chen, "BandNet: A Neural Network-based, Multi-Instrument Beatles-Style MIDI Music Composition Machine," in *ISMIR*, 2019.

[10] S. Dai, X. Ma, Y. Wang, and R. B. Dannenberg, "Personalized Popular Music Generation Using Imitation and Structure," *arXiv preprint arXiv:2105.04709*, 2021.

[11] S.-L. Wu and Y.-H. Yang, "MuseMorphose: Full-Song and Fine-Grained Music Style Transfer with One Transformer VAE," *arXiv preprint arXiv:2105.04090*, 2021.

[12] J. Zhao and G. Xia, "AccoMontage: Accompaniment Arrangement via Phrase Selection and Style Transfer," in *ISMIR*, 2021.

[13] S. Dai, Z. Jin, C. Gomes, and R. B. Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," in *ISMIR*, 2021.

[14] X. Zhang, J. Zhang, Y. Qiu, L. Wang, and J. Zhou, "Structure-Enhanced Pop Music Generation via Harmony-Aware Learning," *arXiv preprint arXiv:2109.06441*, 2021.

[15] J. Ens and P. Pasquier, "MMM: Exploring Conditional Multi-Track Music Generation with the Transformer," *arXiv preprint arXiv:2008.06048*, 2020.

[16] S.-L. Wu and Y.-H. Yang, "The Jazz Transformer on the Front Line: Exploring the Shortcomings of AI-composed Music through Quantitative Measures," in *ISMIR*, 2020.

[17] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, G. Bin, and G. Xia, "POP909: A Pop-song Dataset for Music Arrangement Generation," in *ISMIR*, 2020.

[18] S. Dai, H. Zhang, and R. B. Dannenberg, "Automatic Analysis and Influence of Hierarchical Structure on Melody, Rhythm and Harmony in Popular Music," in *CSMC-MuMe*, 2020.

[19] M. Müller, P. Grosche, and N. Jiang, "A Segment-Based Fitness Measure for Capturing Repetitive Structures of Music Recordings." in *ISMIR*, 2011.

[20] M. Müller and N. Jiang, "A Scape Plot Representation for Visualizing Repetitive Structures of Music Recordings," in *ISMIR*, 2012.

[21] O. Nieto, G. J. Mysore, C. i Wang, J. B. L. Smith, J. Schlüter, T. Grill, and B. McFee, "Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications," *Trans. Int. Soc. Music. Inf. Retr.*, vol. 3, pp. 246–263, 2020.