Vol.2024-MUS-141 No.3
2024/8/26

音楽生成における時刻と音高相対性の重要性

稲葉 達郎[1,a)]　吉井 和佳[1,b)]　中村 栄太[2,c)]

概要：本稿では, 音楽生成において Transformer を効果的に利用する方法について実験的調査を行う．通常，Transformer に基づく音楽生成では，楽譜をイベントあるいは音符を基本単位とするトークン系列に変換して処理する必要がある．しかし，これらの系列では, 音楽の時刻・音高シフトに対する相同性に加え，時刻・音高方向における小節・オクターブ単位の循環性は明示的に表現されていない．そのため，単純に自己注意機構を用いても，音楽特有のリズムやハーモニーの構造を適切に学習できていない可能性がある．この問題に対して，本研究では，時刻と音高の（小節単位とオクターブ単位の）循環性を考慮したトークン間の相対的時刻・音高距離をエンコードし，自己注意機構に用いる手法を提案する．繰り返し構造が比較的明確なポピュラー音楽のデータセット POP909 を用いて，与えられた楽譜の後続部の予測実験を行い，提案法は予測性能を改善することを確認した．また，被験者による主観評価により，生成される音楽の繰り返し構造や一貫性の点で，イベント単位の楽譜表現が Transformer に適していることが分かった．

1. はじめに

Transformer [1] は，自然言語処理（NLP）において提案された系列間学習モデルであり，音楽情報検索（MIR）[2] における生成タスクでも利用されている．Transformer で音楽生成を行うには，楽譜をトークン系列として表現する必要がある．イベント単位表現（例：MIDI [3]，REMI [4]）では各トークンはオンセット時刻か，音高，音価（あるいはオフセット時刻）のいずれかを表すイベントに対応する．音符単位表現（例：OctapleMIDI [5]，MuMIDI [6]）では，各トークンは各音符に対応し，複数イベントをまとめた組として表現される．

音符間の相対的な時刻と音高の関係は，従来の表現方法（図 1）では明示的に表現されていない．異なる絶対時刻でも同じ時刻間隔を持っている音符集合は同じ音楽と認識することができ，時刻シフトに対する音楽の意味的相同性を形成する上で相対的時刻は重要である．また，移調された音楽（異なる絶対音高でも同じ音高間隔を持つ音楽）も同じ音楽として認識できるため，相対的音高においても相同性の概念が存在する．この問題意識のもと，音符の時刻と音高の相対距離を正弦波エンコーディングを通じて自己注意機構に組み込む相対インデックス音高オンセット (Relative Index, Pitch, and Onset; RIPO) 注意機構 [7] が提案されている．

1 京都大学
2 九州大学
a) inaba@sap.ist.i.kyoto-u.ac.jp
b) yoshii.kazuyoshi.3r@kyoto-u.ac.jp
c) nakamura@inf.kyushu-u.ac.jp

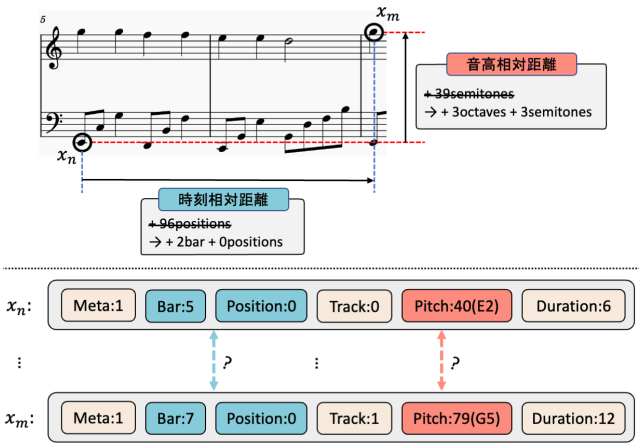

図 1 音符ベースの楽譜表現．音高 40 (E2) のトークン x_n と音高 79 (G5) のトークン x_m に関して，音高距離 39 (3 オクターブ 3 半音) という情報は明示的には表現されていない．

音符間の時刻と音高の相対関係に加え，これらに内在する繰り返し構造と音高の循環性の概念も音楽において重要である．時刻方向において小節は基本単位と見なすことができ，特定小節（1, 2, 4, 8, ... 小節）間隔ごとに似たテーマが演奏される音楽は数多く存在する．同様に，音高方向においてオクターブは基本単位であり，1 オクターブ高い音は元の音の 2 倍の振動数を持ち，似た響きを持つ．楽曲構造を理解する上で，音楽がこれらの基本単位に基づいて作曲されていることを把握することは重要である．例えば，図 1 では，x_n と x_m が 96 時刻単位ではなく 2 小節離れていると解釈し，39 半音ではなく 3 オクターブと 3 つの半音（短 3 度）離れていると解釈する方が効果的に音楽構造を

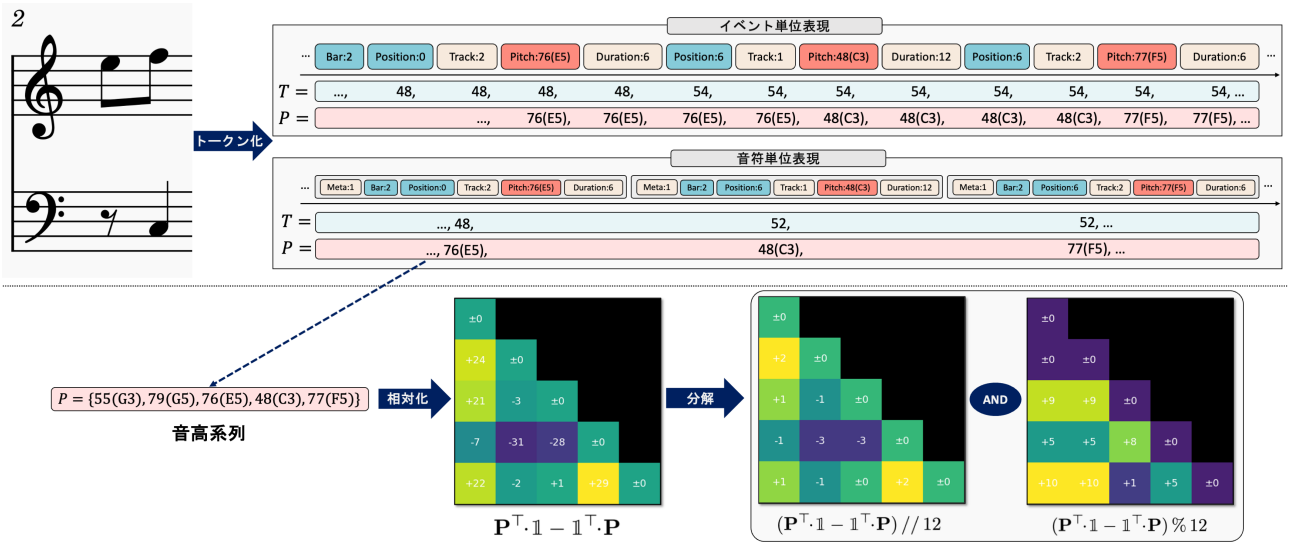

図 2 イベントベース・音符ベースの楽譜表現における時刻 $\mathbf{T}$ および音高系列 $\mathbf{P}$ と，音高のオクターブ単位での循環性に着目した相対音高距離行列.

捉えている.

本稿では，自己注意の計算において, 時刻と音高の（小節単位とオクターブ単位の）循環性を考慮した音符間の時刻・音高の相対距離のエンコーディングを利用する循環相対注意機構を提案する．提案法は，音楽特有の二次元的な繰り返し構造を効果的に捉え，高い一貫性を持つ音楽の生成を可能にする．繰り返し構造が豊富に含まれるポピュラー音楽のデータセット POP-909 [8] を用いて実験を行い，後続生成タスクにおいて，提案法が従来手法よりもテストデータに類似したサンプルを後続生成すること, さらに，生成されたサンプルは，人間評価において特に一貫性の面で高いスコアを獲得したことを報告する.

2. 関連研究

2.1 トークン系列での楽譜表現法

音楽をトークン系列で表現する様々な手法が提案されている [2]．どの表現方法を選択するかは，モデルの音楽構造を学習および一般化する能力に直接影響を与えるため重要である．イベント単位表現 [3,4] では，各トークンが時刻，音高，音価，楽器等の各種音楽イベントを表す．CP word 表現 [9] では，これらの音楽イベントを特定の役割に基づいて単一のトークンに集約する．この手法がさらに拡張され，各トークンが各音符を表す音符単位表現 [5,6] が提案され，系列長効率が向上した.

2.2 音楽構造のモデル化

音楽特有の時刻と音高の構造をモデル化し，ニューラルネットワークに音楽構造を効果的に学習させる方法もいくつか存在する．Chuan ら [10] は，Tonnetz 理論に基づいて音高間の関係を捉える方法を提案した．音高間の関係を二次元で表現し，畳み込みニューラルネットワーク (CNN) と長短期記憶ネットワーク (LSTM) により音楽生成を行なった．Music Transformer [11] では，音楽における相対情報の重要性が主張され，トークン間の相対位置を注意機構に組み込む相対注意機構が提案された．このアプローチは，Guo ら [7] により，音符間の時刻と音高の相対距離をサイン波エンコーディングで自己注意の計算に組み込む RIPO 注意機構へと拡張された．最近では，Agarwal ら [12] により，各タイムステップ間のテンポ，セクション，コード，音高（メロディのみ）の 4 種類の相対関係を自己注意に組み込む手法が提案されている.

3. 提案法

本章では，音楽のトークン系列方法を定義し，既存の相対注意機構と RIPO 注意機構および提案法の循環相対注意機構について説明する.

3.1 トークン系列の定義

音楽をイベント単位（3.1.1 節）あるいは音符単位（3.1.2 節）のトークン系列 $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_L)$ として表現する．簡単のため，両表現においてパフォーマンスに関連する要素（例: テンポや音量）は省略した．また，音楽は時刻解像度が四分音符あたり 12 ステップとなるように量子化を行っていると仮定する.

3.1.1 イベント単位表現

REMI+ [13] を参考に, 表 1 に示すイベントをトークンとして扱う．Bar, Position, Track, Pitch トークンに関する k は範囲内の整数全てをとる．一方，Duration トークンは k は, 次式で表される 27 種類の整数のみを取る.

表 1 イベント単位表現におけるトークン一覧

イベント種類	説明	値の範囲
BOS	曲の開始を表す	
EOS	曲の終了を表す	
Bar:k	k 小節目の開始を表す	$1 \leq k \leq 16$
Position:k	音符が小節頭から k の位置にある	$0 \leq k \leq 47$
Track:k	音符が k トラック目にある	$1 \leq k \leq 3$
Pitch:k	音符の音高が k	$0 \leq k \leq 127$
Duration:k	音符の音価が k	$1 \leq k \leq 96$

$$
\begin{aligned}
k = &\{1, ..., 12\} \\
&\cup \{12 + 3i \mid i \in [1, 4]\} \cup \{12 + 4i \mid i \in [1, 3]\} \\
&\cup \{24 + 6i \mid i \in [1, 4]\} \cup \{48 + 12i \mid i \in [1, 4]\} \quad (1)
\end{aligned}
$$

四分音符まではタイムステップ，二分音符までは十六分音符と一拍三連符，全音符までは八分音符，二全音符までは四分音符の音価がそれぞれ表現できる．REMI+と異なり，拍子，テンポ，コード，および音量のイベントが省略されており，本研究では全部で $1+1+16+8+3+128+27 = 184$ 種類のトークンを使用した．

図 1 の上部にイベント単位表現の例を示す．トークン系列は BOS トークンで始まり，EOS トークンで終わる．時刻と音高でソートした音符を順に Position，Track，Pitch，Duration の 4 トークンの並びで表現し，これを順にトークン系列に追加する．また，小節が変わった際には Bar トークンを追加する．

また，3.2, 3.3, 3.4 節で使用する位置系列 $\mathbf{I} = \{I_1, ..., I_L\}$ と，時刻系列 $\mathbf{T} = \{T_1, ..., T_L\}$，音高系列 $\mathbf{P} = \{P_1, ..., P_L\}$ を Algorithm 1 で定義する．BarRes は一拍あたりのタイムステップ数を表す．

3.1.2 音符単位表現

本研究では Multitrack Music Transformer [6] を基にした表現を 音符単位表現として使用する．まず，各トークン（音符）は 6 つの変数の組として表現される．

$$
\mathbf{x}_i = \{x_i^{\text{meta}}, x_i^{\text{bar}}, x_i^{\text{position}}, x_i^{\text{track}}, x_i^{\text{pitch}}, x_i^{\text{duration}}\}
$$

これらの 6 つの変数にはそれぞれ順に Meta, Bar, Position, Track, Pitch, Duration トークンが対応し, Meta と Bar トークン以外は全て 3.1.1 節におけるイベント種類の定義と同じである．Meta トークンは全部で三種類存在し，それぞれ曲の開始 (0)，音符 (1)，曲の終了 (2) を表す．Bar トークンは音符の存在する小節を表し，値の範囲はイベント単位と同様に 1-16 である．トークンをエンコードする際には，各変数をそれぞれ別々に線形変換層により変換し，その和がトークンの潜在表現となる．逆にトークンをデコードする際には，出力の潜在表現に対し 6 種類の線形変換層により各変数を別々にデコードする．

図 1 の上部に音符単位表現の例を示す．音符を全てトークンに変換したのち，時刻と音高によりソートしたトークン系列に対し，曲の開始を表すトークン (Meta:0) を最初にと曲の最後を表すトークン (Meta:2) を最後にそれぞれ追加する．また，音符単位表現の位置系列と，時刻系列，音高系列を次のように定義する．

Algorithm 1 イベント単位 $\mathbf{I}, \mathbf{T}, \mathbf{P}$

```
cbar ← None #current bar
cpos ← None #current position
cpit ← None #current pitch
I, T, P ← [], [], []
for idx, event ∈ enumerate(sequence) do
    if type of event is Bar then
        cbar ← Bar value of event
    end if
    if type of event is Position then
        cpos ← Position value of event
    end if
    if type of event is Pitch then
        cpit ← Pitch value of event
    end if
    I.append(idx)
    T.append(cbar × BarRes + cpos)
    P.append(cpit)
end for
```

$$
\mathbf{I} = \{1, ..., L\} \quad (2)
$$

$$
\mathbf{T} = \{x_i^{\text{bar}} \times \text{BarRes} + x_i^{\text{position}}\}_{i=1}^{L} \quad (3)
$$

$$
\mathbf{P} = \{x_i^{\text{pitch}}\}_{i=1}^{L} \quad (4)
$$

3.2 相対注意機構

注意への入力は，各要素が D 次元の長さ L のベクトル $\mathbf{X} = \{X_1, ..., X_L\} \in \mathbb{R}^{L \times D}$ である．入力系列は，クエリ $Q = \mathbf{X}W^Q$，キー $K = \mathbf{X}W^K$，およびバリュー $V = \mathbf{X}W^V$ に変換される．ただし，$W_Q, W_K, W_V \in \mathbf{R}^{D \times D_h}$ である．ここで，D_h は各注意機構の各ヘッドにおける次元数を表す．本来の Transformer では，マルチヘッド注意機構が採用されており，様々なサブスペースの情報に同時に注意を向けることが可能である．本研究でも，このマルチヘッド注意機構を活用しているが，以降の計算では明確さのために一ヘッドのみの計算に焦点を当てている．

Transformer の注意機構の出力は，次式で定義される．

$$
\text{Attn} = \text{Softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V \quad (5)
$$

一方，相対注意機構 [11, 14] では，系列内のトークン間の相対距離距離に基づいた位置埋め込みが注意機構に組み込まれる．

$$
\mathbf{R}_{\text{idx}} = \text{LPE}(\mathbf{I}^\top \cdot \mathbb{1} - \mathbb{1}^\top \cdot \mathbf{I}) \quad (6)
$$

$$
S_{\text{rel}}^{\text{idx}} = Q\mathbf{R}_{\text{idx}}^T \quad (7)
$$

$$
\text{RelAttn} = \text{Softmax}\left(\frac{QK^T + \alpha S_{\text{rel}}^{\text{idx}}}{\sqrt{D}}\right)V \quad (8)
$$

ただし，$\mathbb{1}$ は長さ L のすべての要素が 1 である行ベクト

ルである．相対位置行列 $\mathbf{I}^\top \cdot \mathbb{1} - \mathbb{1}^\top \cdot \mathbf{I}$ が学習可能なエンコーディング LPE により埋め込まれ，その埋め込み $\mathbf{R}_{\text{idx}}$ とクエリ Q の積により計算されたロジット $S_{\text{rel}}^{\text{idx}}$ が QK^T と足し合わされる．

3.3 RIPO 注意機構

RIPO 注意機構 [7] では，相対位置埋め込みに加え，音楽内における音符間の相対時刻と相対音高も考慮する．

$$\mathbf{R}_{\text{time}} = \text{SPE}(\mathbf{T}^\top \cdot \mathbb{1} - \mathbb{1}^\top \cdot \mathbf{T}) \tag{9}$$

$$\mathbf{R}_{\text{pitch}} = \text{SPE}(\mathbf{P}^\top \cdot \mathbb{1} - \mathbb{1}^\top \cdot \mathbf{P}) \tag{10}$$

$$S_{\text{rel}}^{\text{t}} = Q\mathbf{R}_{\text{time}}^T \tag{11}$$

$$S_{\text{rel}}^{\text{p}} = Q\mathbf{R}_{\text{pitch}}^T \tag{12}$$

$$S_{\text{rel}} = S_{\text{rel}}^{\text{idx}} + S_{\text{rel}}^{\text{t}} + S_{\text{rel}}^{\text{p}} \tag{13}$$

$$\text{RIPOAttn} = \text{Softmax}\left(\frac{QK^T + \alpha S_{\text{rel}}}{\sqrt{D}}\right)V \tag{14}$$

相対時刻行列 $\mathbf{T}^\top \cdot \mathbb{1} - \mathbb{1}^\top \cdot \mathbf{T}$ と相対音高行列 $\mathbf{P}^\top \cdot \mathbb{1} - \mathbb{1}^\top \cdot \mathbf{P}$ は，正弦波エンコーディング SPE を使用して埋め込まれ，これら埋め込み $\mathbf{R}_{\text{t}}, \mathbf{R}_{\text{p}}$ とクエリ Q の積が相対時刻と相対音高のロジット $S_{\text{rel}}^{\text{t}}, S_{\text{rel}}^{\text{p}}$ として注意機構の計算に用いる．

3.4 循環相対注意機構

本節では，提案法の循環相対注意機構を説明する．時刻と音高の相対距離に内在する基本単位を考慮することで，RIPO 注意機構では捉えれていなかった繰り返し構造と音高の循環的な性質をモデリングする．まず，相対時刻距離行列を小節単位とその余りに分解し，学習可能な位置エンコーディングを使用してそれぞれの行列を埋め込む．

$$\mathbf{R}_{\text{bar}} = \text{LPE}((\mathbf{T}^\top \cdot \mathbb{1} - \mathbb{1}^\top \cdot \mathbf{T}) \,//\, \text{BarRes}) \tag{15}$$

$$\mathbf{R}_{\text{position}} = \text{LPE}((\mathbf{T}^\top \cdot \mathbb{1} - \mathbb{1}^\top \cdot \mathbf{T}) \,\%\, \text{BarRes}) \tag{16}$$

同様に，相対音高距離行列をオクターブ単位とその余りに分解し，学習可能な位置エンコーディングを使用してそれぞれの行列を埋め込む．

$$\mathbf{R}_{\text{octave}} = \text{LPE}((\mathbf{P}^\top \cdot \mathbb{1} - \mathbb{1}^\top \cdot \mathbf{P}) \,//\, 12) \tag{17}$$

$$\mathbf{R}_{\text{semitone}} = \text{LPE}(\mathbf{P}^\top \cdot \mathbb{1} - \mathbb{1}^\top \cdot \mathbf{P}) \,\%\, 12) \tag{18}$$

ここで，正弦波位置エンコーディング（SPE）を使用しない理由は，相対距離の近さが必ずしも強い関係を示すわけではないからである．例えば，[15] で指摘されているように，似たような音楽的構造は特定小節前 (1, 2, 4, 8, 16... 小節) に現れやすい傾向がある．また，音高距離に対する和声的特徴は複雑であり，各和音が独自の共鳴を持っている．これらの相対距離ごとの特徴を別々に捉えるために学習可能エンコーディング LPE を使用した．各種埋め込みから時刻と音高のロジットは次式のいずれかで与えられる．

$$\begin{cases} S_{\text{rel}}^{\text{t}} = Q\left(\mathbf{R}_{\text{bar}} \odot \mathbf{R}_{\text{position}}\right)^T \\ S_{\text{rel}}^{\text{p}} = Q\left(\mathbf{R}_{\text{octave}} \odot \mathbf{R}_{\text{semitone}}\right)^T \end{cases} \tag{19}$$

$$\begin{cases} S_{\text{rel}}^{\text{t}} = Q\left(\mathbf{R}_{\text{bar}} + \mathbf{R}_{\text{position}}\right)^T \\ S_{\text{rel}}^{\text{p}} = Q\left(\mathbf{R}_{\text{octave}} + \mathbf{R}_{\text{semitone}}\right)^T \end{cases} \tag{20}$$

ここで，式 (19) を循環相対-H，式 (20) を 循環相対-S とする．

4. 実験的評価

本章では，後続楽譜の予測実験の客観評価及び主観評価結果について述べ，提案法の有効性について検証する．

4.1 実験条件

繰り返し構造が豊富に含まれるポピュラー楽曲のデータセット POP909 [8] を利用した．ビートの注釈が不一致な楽曲を除外し，残った 896 曲を 10 %を検証用，10 %をテスト用，残りの 80 %を訓練用に割り振った．各データが幅 16 小節となるように，ストライド 1 小節で各楽曲からデータを作成した．また，この際に 16 小節通して 4/4 拍子のサンプルのみを保持した結果，検証用に 4,749，テスト用に 4,137，訓練用に 35,452 のデータ数となった．また，時刻方向のステップ幅 (Resolution) を四分音符ごとに 12 ステップとなるようにクオンタイズを行なった．これらのデータの前処理には MusPy [16] を使用した．訓練中には，各データを-6 から+5 半音の範囲でランダムに移調するデータ拡張を行った．モデルは，従来手法，提案法共にモデル次元 256，層数 4，自己注意機構のヘッド数 8，ドロップアウト率 0.2 のデコーダーのみの Transformer を使用した．学習時は 1,000 ステップごとに検証を行い，200,000 ステップ後あるいは 20 回の検証で損失の改善がない場合に学習を終了した．バッチサイズは 8 に固定し，ウォームアップステップは 10,000，ピーク学習率は 2e-5 とした．また，ハイパーパラメータの α は 0.1，γ は 0.2 にそれぞれ設定した．

比較対象として，Transformer の元論文で提案された注意機構 [1] と，Music Transformer で提案された相対注意機構 [11]，RIPO 注意機構 [7] を使用した．

4.2 客観評価

後続生成タスクにおいて，モデルが元の音楽にどれだけ近い音楽を生成できるかを客観評価指標により評価する．16 小節からなる各テストデータの最初の 15 小節をモデルに与え，続きの 1 小節を生成させる．生成された 1 小節と実際の 1 小節 (Ground Truth, GT) との類似性を，五種類の基準で評価する．

- NoteF1 ($F1_{\text{note}}$) または OnsetF1 [17] は，生成されたサンプルと GT が音符単位でどれだけ一致しているかを F1 値により測定する．音符の開始タイミング，音

表 2 後続生成タスクにおける客観評価結果

No.	表現方法	注意機構	Params	Time (ms/note)	損失	$F1_{\text{note}}$	$F1_{\text{pr}}$	ND	CS	GS	PRS
1		標準 [1]	4.32M	8.46	0.979	0.174	0.239	0.908	0.620	0.846	0.955
2		相対 [11]	4.84M	9.73	0.937	0.218	0.291	0.903	0.650	0.848	0.955
3	イベント単位	RIPO [7]	4.85M	18.3	0.904	0.233	0.305	0.908	0.660	0.855	0.956
4		循環相対-S	4.88M	22.5	0.876	0.268	0.341	0.909	0.673	0.855	0.957
5		循環相対-H	4.88M	20.8	**0.856**	**0.293**	**0.361**	**0.914**	**0.685**	**0.858**	**0.958**
6		標準 [1]	3.53M	4.29	4.42	0.188	0.267	0.873	0.619	0.784	0.941
7		相対 [11]	3.66M	5.79	4.38	0.186	0.271	0.880	0.628	**0.798**	**0.946**
8	音符単位	RIPO [7]	3.67M	5.93	4.40	0.180	0.257	0.878	0.612	0.786	0.942
9		循環相対-S	3.70M	8.58	4.37	0.192	0.273	0.877	0.623	0.785	0.943
10		循環相対-H	3.70M	6.54	**4.29**	**0.215**	**0.294**	**0.881**	**0.632**	0.787	0.944

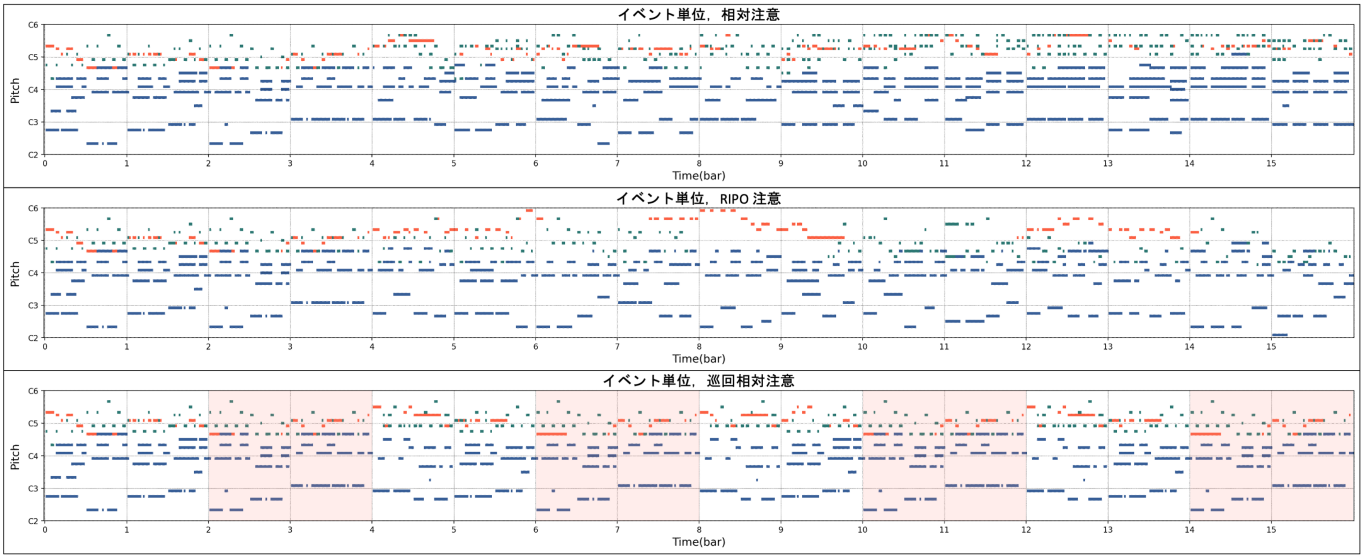

図 3 イベント単位表現の相対，RIPO，循環相対注意機構により生成されたサンプル．最初の 4 小節を与え，続きの 12 小節を生成している．相対注意機構と RIPO 注意機構が完全に新しいフレーズを生成する一方，循環注意機構は入力として与えたフレーズに似たピースを繰り返し生成している（赤く塗りつぶされた部分が繰り返し構造）．

高，トラックが全て一致する場合に真陽性としてカウントする．

- PianorollF1 ($F1_{\text{pr}}$) または FrameF1 [17] は，生成されたサンプルと GT がピアノロール表現上においてどれだけ一致しているかを各タイムステップ・音高ごとに判定し，F1 値により測定する．
- Chroma Similarity (CS) [18] は，生成されたサンプルと GT のクロマベクトル [19] のコサイン類似度である．クロマベクトルは，特定の時刻範囲内での 12 の音高クラス (C, C#, ..., B) の各クラスに対するオンセットの数を表す 12 次元のベクトルであり，本実験では，半小節ごと（24 タイムステップ）にコサイン類似度を計算し，その平均を計算する．
- Grooving Similarity (GS) [18,20] は，生成された音楽と GT のグルーヴィングベクトルのコサイン類似度である．各グルーヴィングベクトルは，小節内の各タイムステップでのオンセットの数を表し，本実験では 48 次元のベクトルとなる．
- Pitch Range Similarity (PRS) は，生成された音楽と GT の小節内の音高範囲の類似性である．

$$\text{PRS} = 1 - \frac{|\text{PR}_{\text{gen}} - \text{PR}_{\text{gt}}|}{128} \tag{21}$$

ここで，PR_{gen} と PR_{gt} は，それぞれ生成された音楽と GT の音高範囲（最高音と最低音の音高差）をそれぞれ表す．

表 2 に，後続生成タスクにおける客観評価結果を示す．提案法（No. 5, 10）は，イベント単位と 音符単位表現の両表現においてほとんどの客観的指標で従来手法モデルよりも優れた性能を示した．すなわち，提案法が従来手法に比べ，テストデータに類似したサンプルを後続生成していることを確認した．

表 3　リスニングテストにおける主観評価結果

手法		主観評価		
表現方法	注意機構	一貫性	音楽性	総合点
Ground Truth		3.93	4.17	4.03
イベント単位	相対 [11]	2.93	2.69	2.79
	RIPO [7]	3.0	3.21	3.03
	循環相対-H	**4.31**	**3.41**	**3.69**
音符単位	相対 [11]	2.28	2.69	2.45
	RIPO [7]	**2.69**	**2.90**	**2.90**
	循環相対-H	2.10	2.52	2.41

4.3　主観評価

提案法で生成した音楽サンプル自体を評価するために，リスニングテストを実施する．6 つのモデル（イベント単位と音符単位の相対注意と，RIPO 注意，循環注意-H）により生成したサンプルに加え，Ground Truth の 7 種類を評価する．生成サンプルは，テストデータの最初の 4 小節だけをモデルに与え，その続き 12 小節を生成することで作成する．各評価者は 7 種類のサンプルを聞き，それらを一貫性，音楽性，全体的な質の 3 つの観点で 1 から 5 の五段階評価を行う．各サンプルセットごとに 3 人が評価するように，30 人の評価者を募り，10 のサンプルセットで評価を行う．

表 3 にリスニングテストの結果を示す．イベント単位表現における提案法（循環相対-H）は，特に一貫性において他の手法よりも高い評価を得た．生成されたサンプルを分析したところ，イベント単位表現の循環相対-H は繰り返し構造を多く生成する傾向があった．図 3 に，イベント単位表現の相対，RIPO，循環相対-H により生成されたサンプルの例をそれぞれ示す．相対と RIPO は常に新しいフレーズを生成しているのに比べ，循環相対-H は与えられた最初の 4 小節の中にあるフレーズを繰り返し生成することで，ポピュラー音楽の特徴をうまく捉えている．繰り返し構造を多く含む音楽が人間の評価者により一貫性の面で特に高い評価を得たと考えられる．提案法により，モデルは音楽特有の繰り返し構造の特徴をよく掴み，一貫した音楽を生成できるようになった．

5.　おわりに

本稿では，時刻と音高の相対距離をそれぞれ小節とオクターブを基に分解し，別々で学習可能エンコーディングにより自己注意機構の計算に組み込む手法を提案した．相対距離を分解して扱うことで，モデルは音楽構造をより効果的に学習し，テストデータをより高い精度で予測できることを客観評価から確認した．また，提案法が繰り返し構造を多く含む一貫性の高いサンプルを生成することもリスニングテストによる主観評価から確認した．

謝辞　本研究は，JST FOREST No. JPMJFR2270 および JSPS 科研費 Nos. 24H00742, 24H00748 の支援を受けた．

参考文献

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is all you need, *Neural Information Processing Systems*, pp. 5998–6008 (2017).

[2] Ji, S., Yang, X. and Luo, J.: A Survey on Deep Learning for Symbolic Music Generation: Representations, Algorithms, Evaluations, and Challenges, *ACM Comput. Surv.*, Vol. 56, No. 1 (2023).

[3] Oore, S., Simon, I., Dieleman, S., Eck, D. and Simonyan, K.: This time with feeling: learning expressive musical performance, *Neural Computing and Applications*, Vol. 32, pp. 955 – 967 (2018).

[4] Huang, Y.-S. and Yang, Y.-H.: Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions, *ACM International Conference on Multimedia*, p. 1180–1188 (2020).

[5] Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T. and Liu, T.-Y.: MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 791–800 (2021).

[6] Dong, H.-W., Chen, K., Dubnov, S., McAuley, J. and Berg-Kirkpatrick, T.: Multitrack Music Transformer, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2023).

[7] Guo, Z., Kang, J. and Herremans, D.: A domain-knowledge-inspired music embedding space and a novel attention mechanism for symbolic music modeling, *AAAI Conference on Artificial Intelligence*, Vol. 37, No. 4, pp. 5070–5077 (2023).

[8] Wang, Z., Chen, K., Jiang, J., Zhang, Y., Xu, M., Dai, S., Bin, G. and Xia, G.: POP909: A Pop-song Dataset for Music Arrangement Generation, *International Society for Music Information Retrieval Conference ISMIR* (2020).

[9] Hsiao, W.-Y., Liu, J.-Y., Yeh, Y.-C. and Yang, Y.-H.: Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs, *AAAI Conference on Artificial Intelligence* (2021).

[10] Chuan, C.-H. and Herremans, D.: Modeling Temporal Tonal Relations in Polyphonic Music Through Deep Networks With a Novel Image-Based Representation, *PAAAI Conference on Artificial Intelligence*, Vol. 32, No. 1 (2018).

[11] Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A. M., Hoffman, M. D., Dinculescu, M. and Eck, D.: Music Transformer, *International Conference on Learning Representations ICLR* (2019).

[12] Agarwal, M., Wang, C. and Richard, G.: Structure-informed Positional Encoding for Music Generation, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2024).

[13] von Rütte, D., Biggio, L., Kilcher, Y. and Hofmann, T.: FIGARO: Controllable Music Generation using Learned and Expert Features, *International Conference on Learning Representations ICLR* (2023).

[14] Shaw, P., Uszkoreit, J. and Vaswani, A.: Self-Attention with Relative Position Representations, *North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468 (2018).

[15] Yu, B., Lu, P., Wang, R., Hu, W., Tan, X., Ye, W., Zhang,

S., Qin, T. and Liu, T.-Y.: Museformer: Transformer with Fine- and Coarse-Grained Attention for Music Generation, *Neural Information Processing Systems* (2022).

[16] Dong, H.-W., Chen, K., McAuley, J. and Berg-Kirkpatrick, T.: MusPy: A Toolkit for Symbolic Music Generation, *International Society for Music Information Retrieval Conference ISMIR* (2020).

[17] Gardner, J. P., Simon, I., Manilow, E., Hawthorne, C. and Engel, J.: MT3: Multi-Task Multitrack Music Transcription, *International Conference on Learning Representations ICLR* (2022).

[18] Wu, S.-L. and Yang, Y.-H.: MuseMorphose: Full-Song and Fine-Grained Piano Music Style Transfer with One Transformer VAE, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023).

[19] Fujishima, T.: Realtime Chord Recognition of Musical Sound : a System Using Common Lisp Music, *International Computer Music Conference ICMC*, pp. 464–467 (1999).

[20] Dixon, S., Gouyon, F. and Widmer, G.: Towards Characterisation of Music via Rhythmic Patterns, *International Society for Music Information Retrieval Conference ISMIR* (2004).