

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, 吉井和佳¹, 中村栄太²

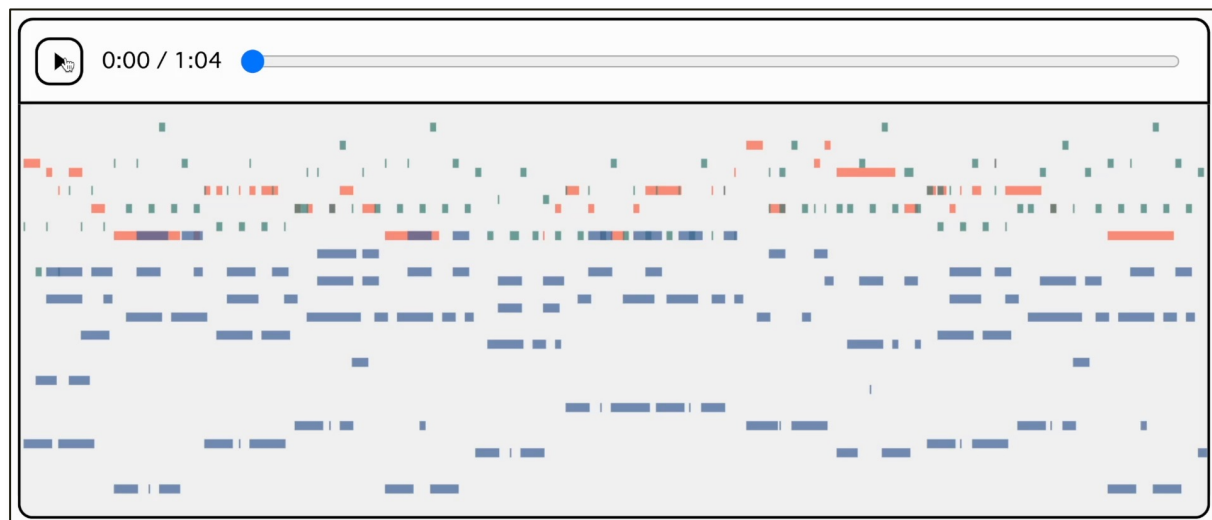
¹ 京都大学 ² 九州大学

2024/8/26, 第141回音楽情報科学研究発表会



研究概要

- 記号音楽（楽譜）生成において Transformer の能力を最大限引き出す手法の模索
- 音符間の時刻と音高の**相対距離**を，小節単位とオクターブ単位の**循環性**を考慮したエンコーディングにより自己注意機構に組み込んだ
- 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生成が可能

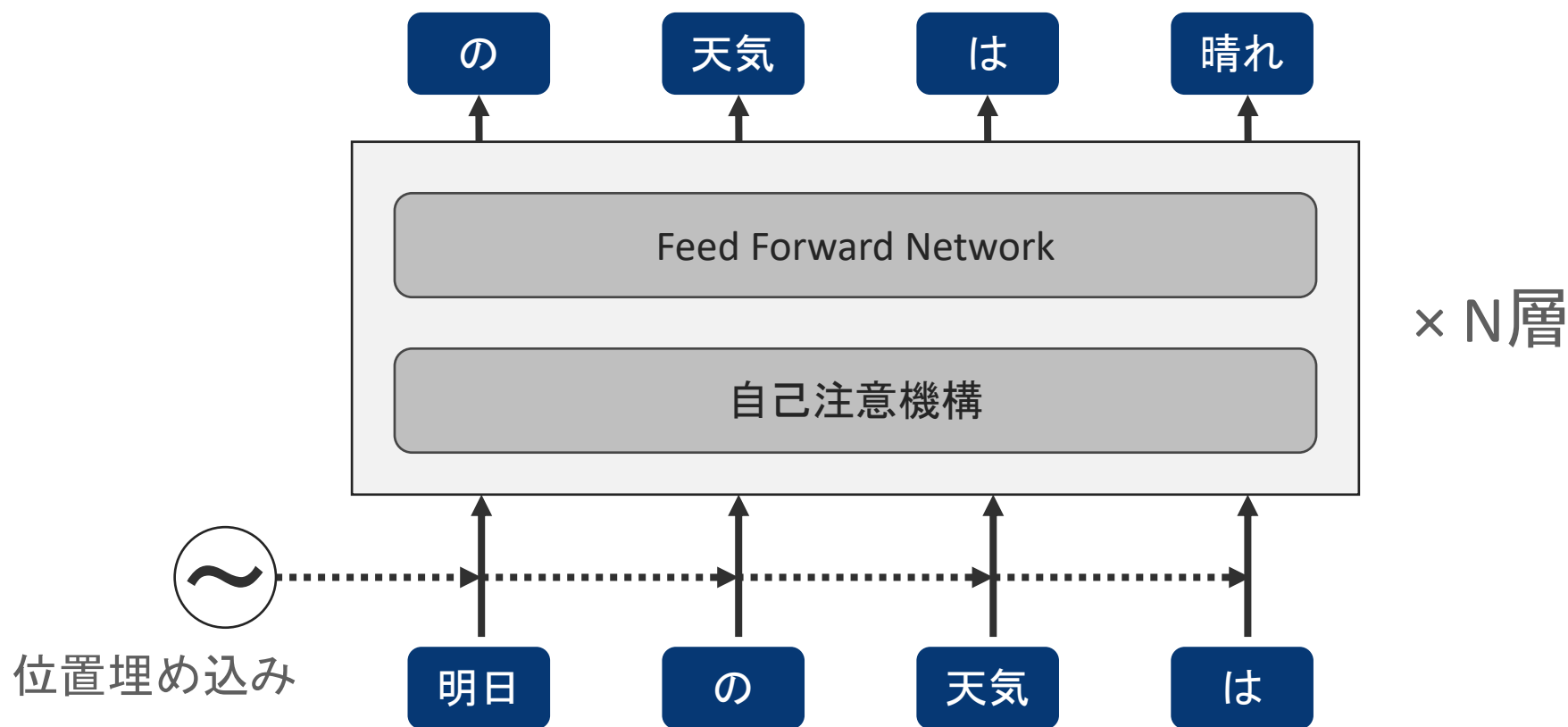


提案手法の生成音楽例

16秒までの4小節をプロンプトとして与え，
その続き12小節を生成させている

デコーダー型 Transformer によるテキスト生成

- 入力したトークンの次のトークンを予測
- 再起的に予測を行うことでテキスト生成が可能



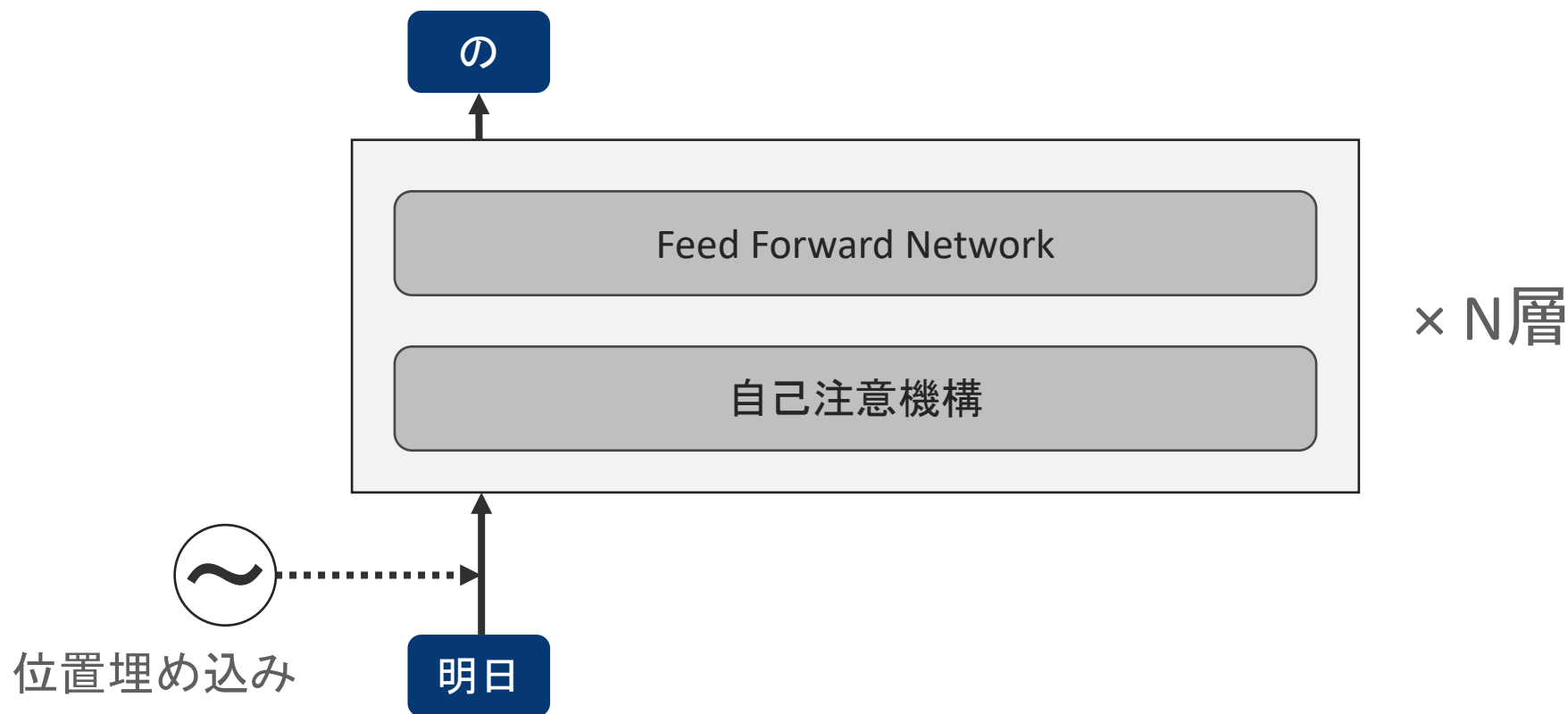
デコーダー型 Transformer によるテキスト生成

- 入力したトークンの次のトークンを予測
- 再起的に予測を行うことでテキスト生成が可能



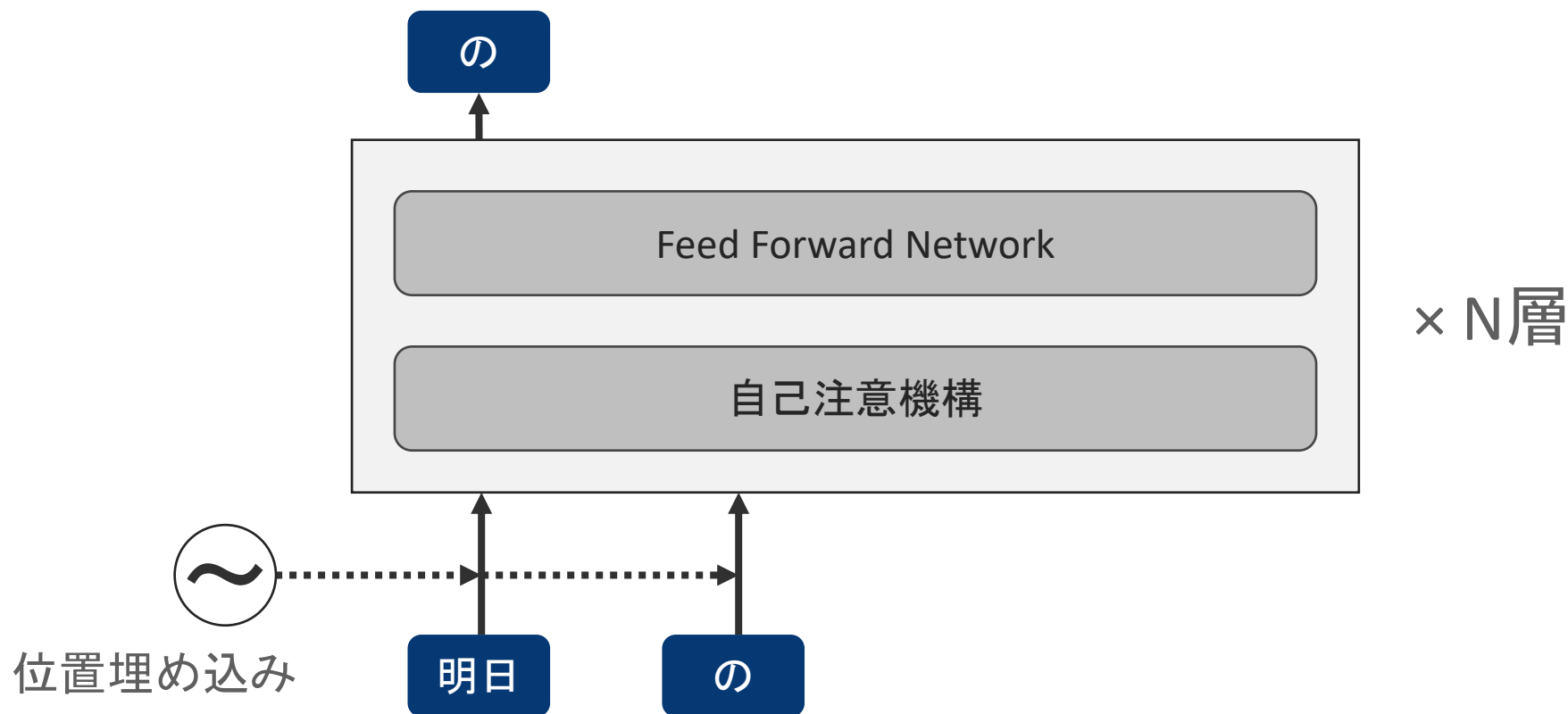
デコーダー型 Transformer によるテキスト生成

- 入力したトークンの次のトークンを予測
- 再起的に予測を行うことでテキスト生成が可能



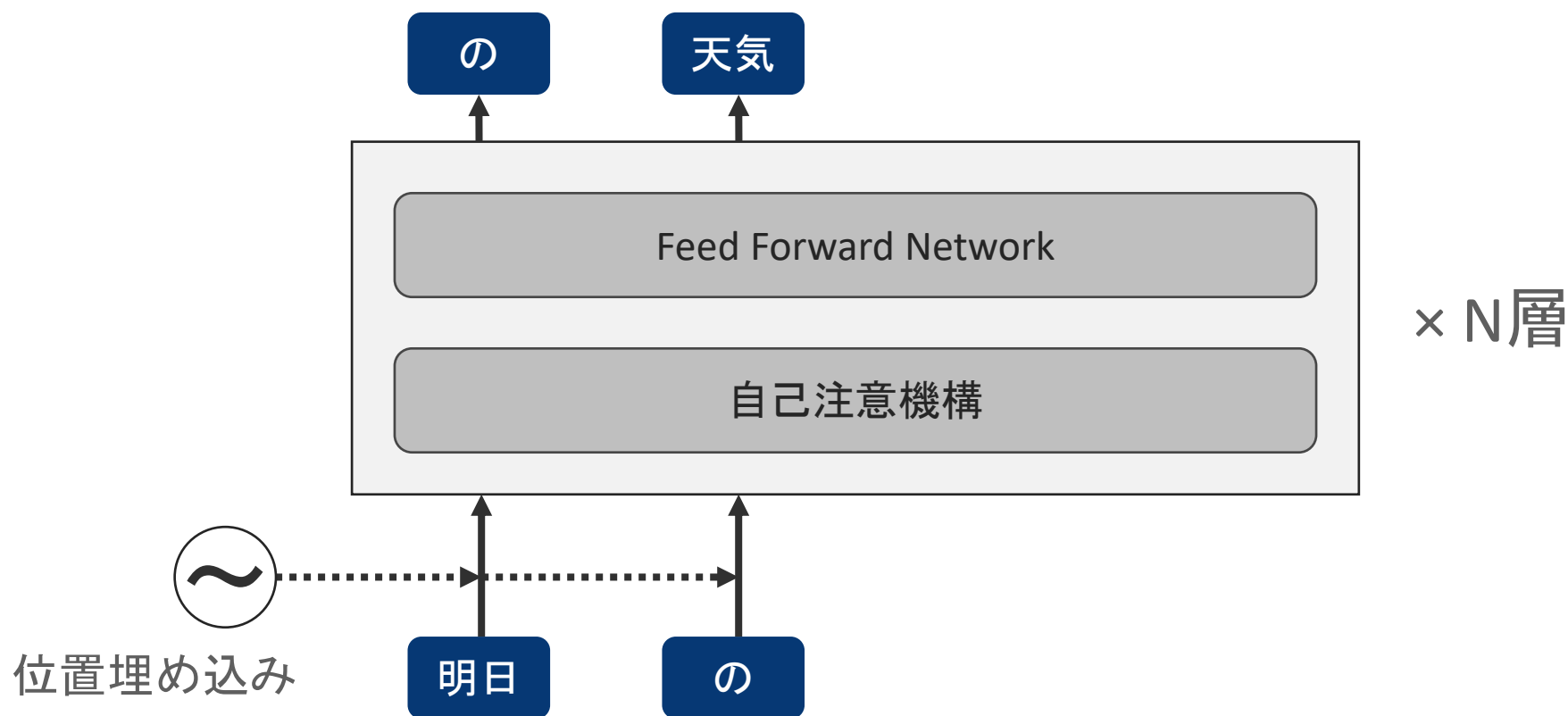
デコーダー型 Transformer によるテキスト生成

- 入力したトークンの次のトークンを予測
- 再起的に予測を行うことでテキスト生成が可能



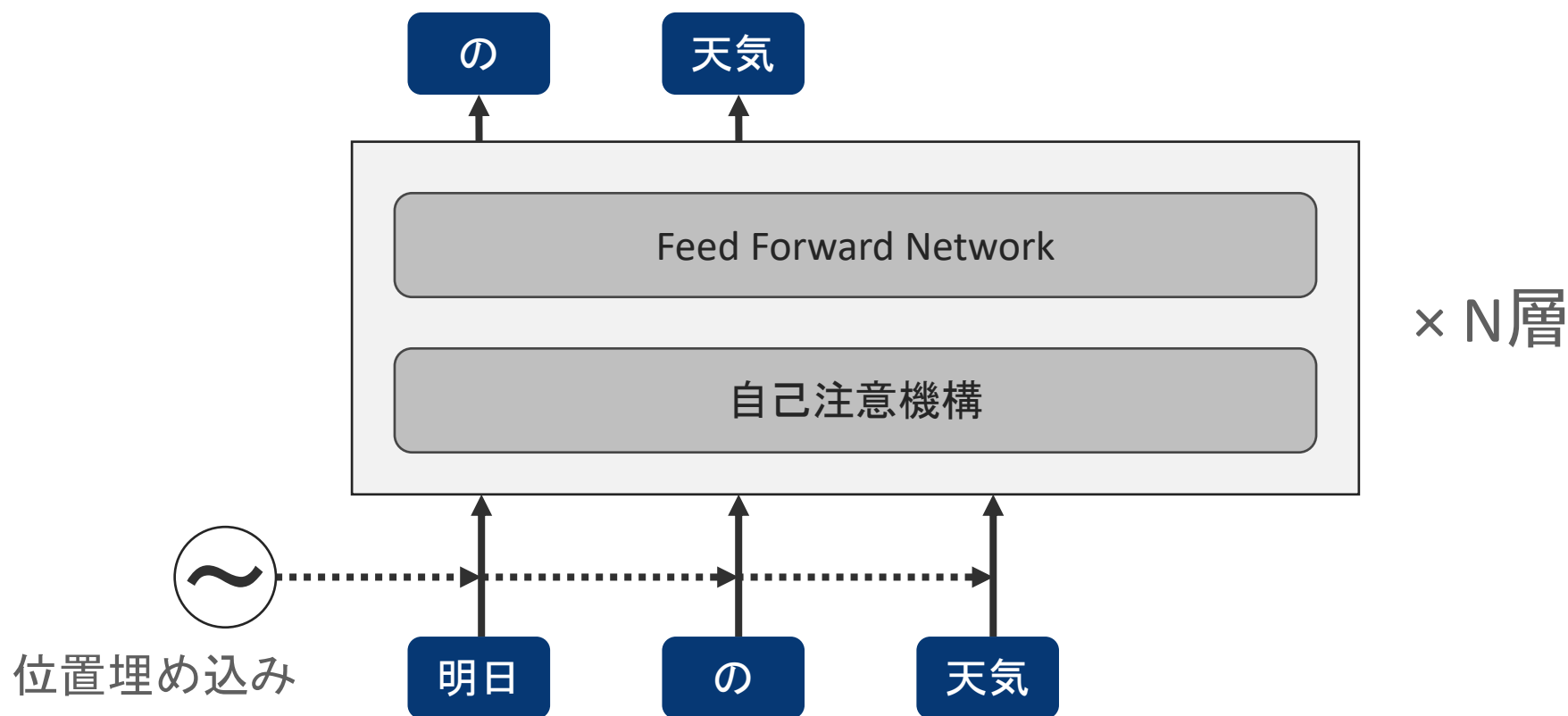
デコーダー型 Transformer によるテキスト生成

- 入力したトークンの次のトークンを予測
- 再起的に予測を行うことでテキスト生成が可能



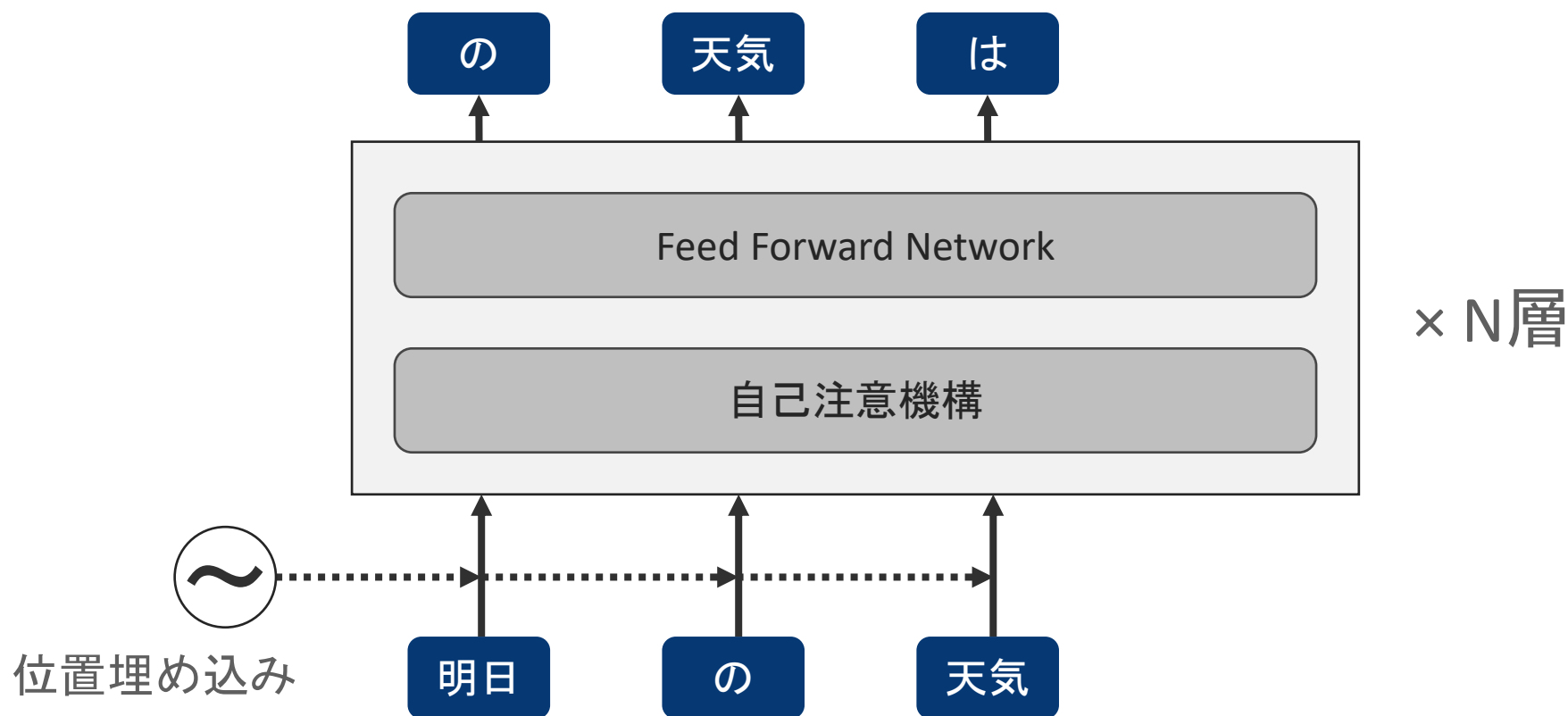
デコーダー型 Transformer によるテキスト生成

- 入力したトークンの次のトークンを予測
- 再起的に予測を行うことでテキスト生成が可能



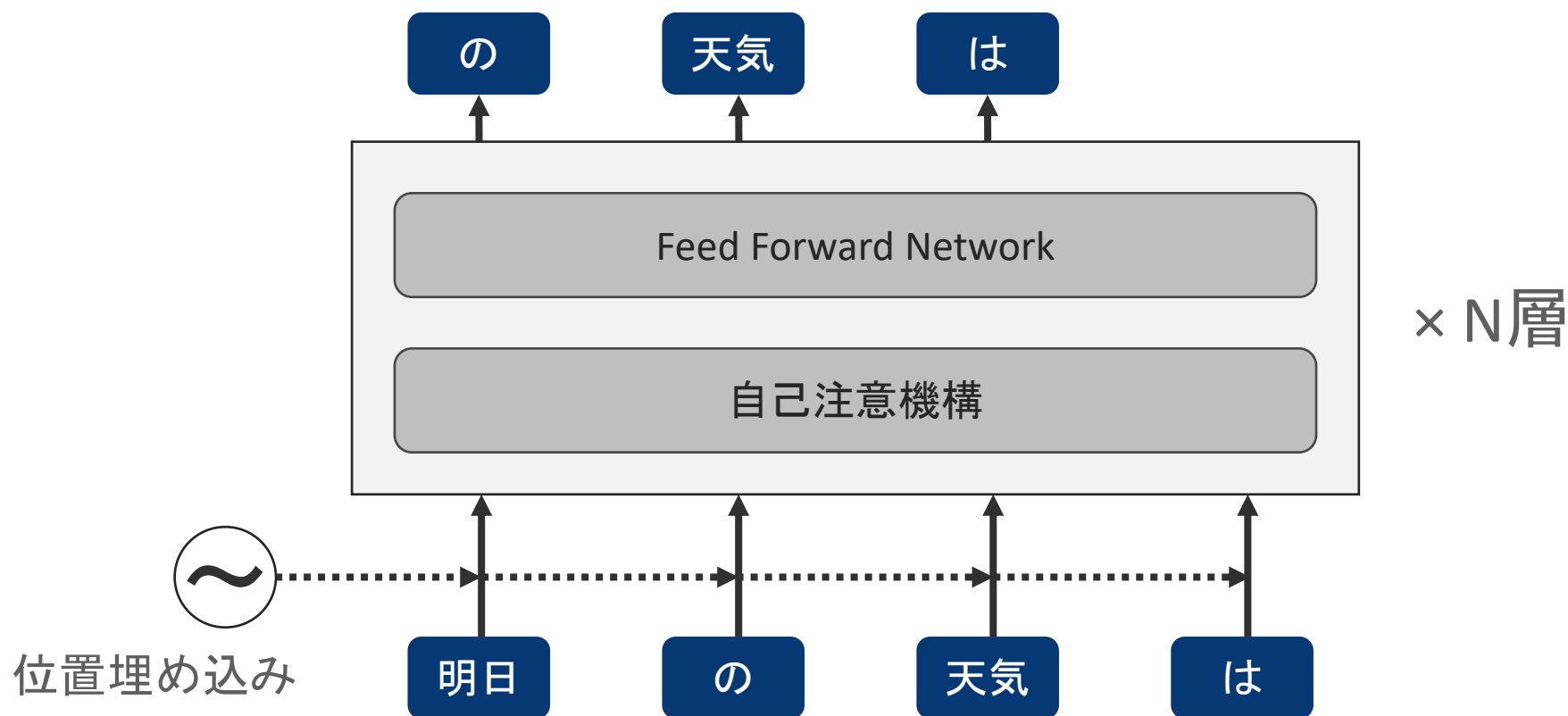
デコーダー型 Transformer によるテキスト生成

- 入力したトークンの次のトークンを予測
- 再起的に予測を行うことでテキスト生成が可能



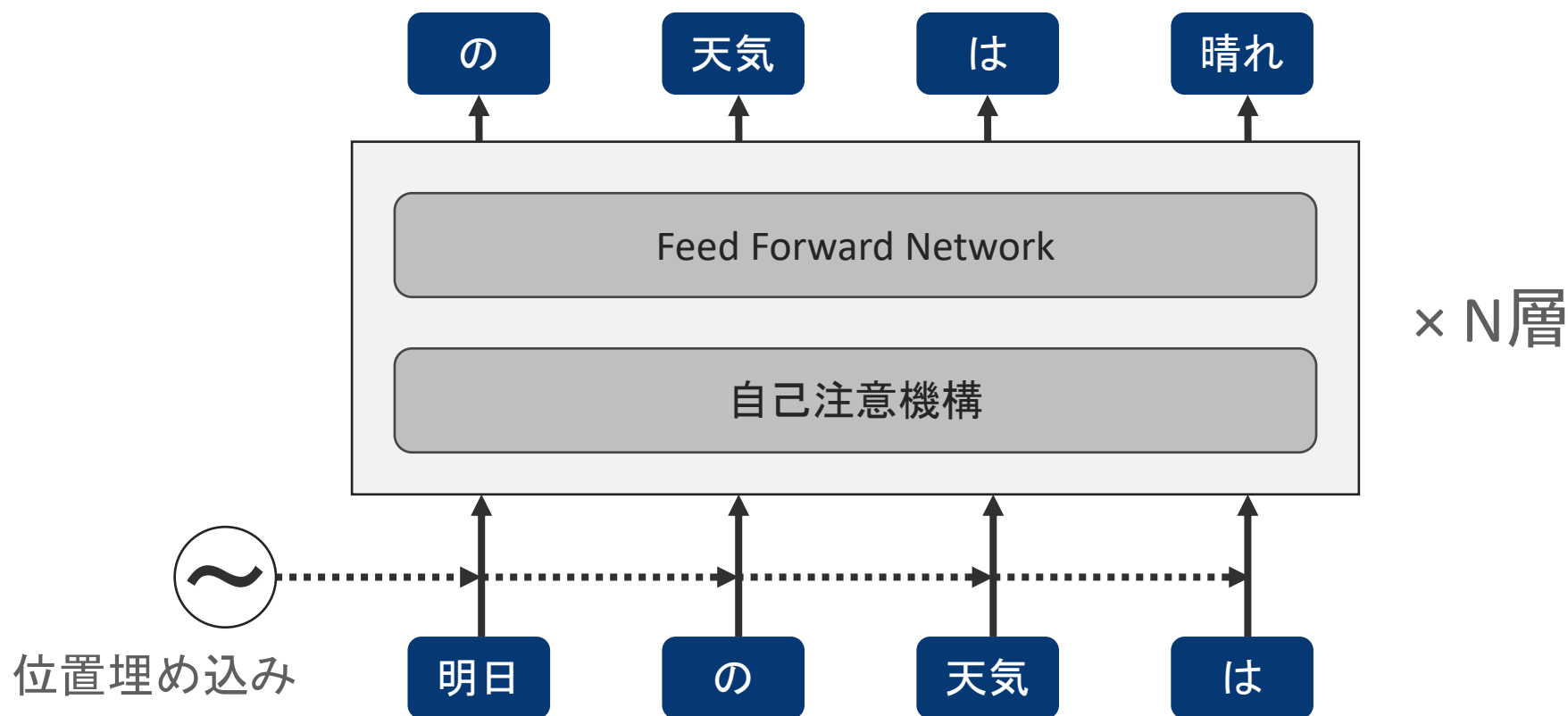
デコーダー型 Transformer によるテキスト生成

- 入力したトークンの次のトークンを予測
- 再起的に予測を行うことでテキスト生成が可能



デコーダー型 Transformer によるテキスト生成

- 入力したトークンの次のトークンを予測
- 再起的に予測を行うことでテキスト生成が可能



デコーダー型 Transformer による音楽生成

- 音楽を Transformer が処理できるトークン列に変換する必要がある

- 音符単位表現
- イベント単位表現

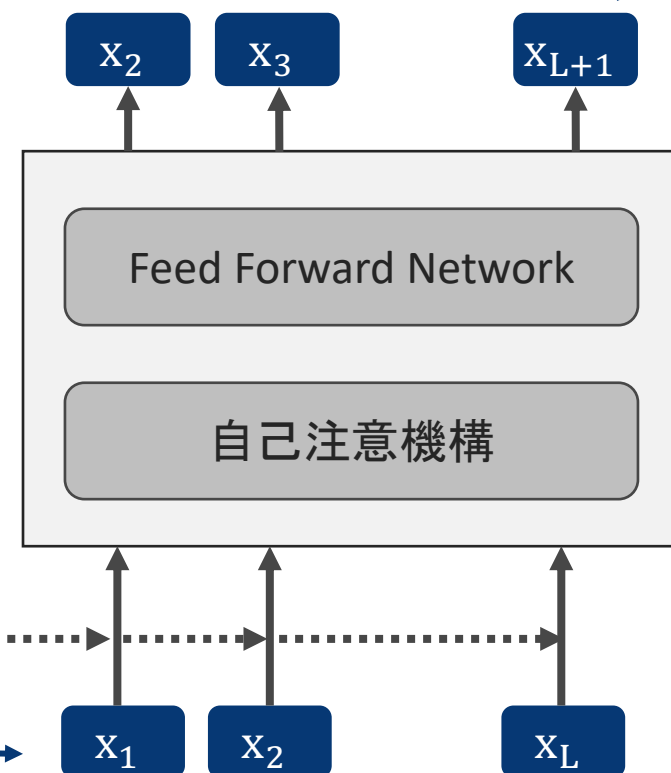
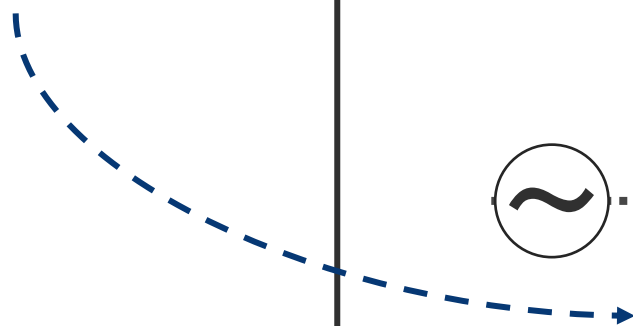


トークン化



???

$\{x_1, x_2, \dots, x_L\}$



続きの音楽
を生成

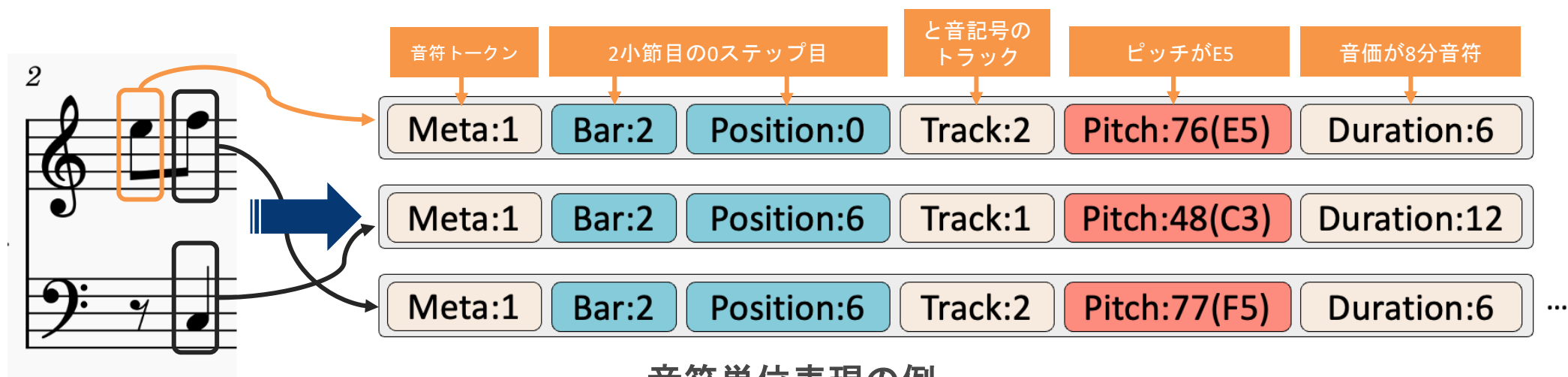
× N層

音符単位表現 [Zeng+'21; Dong+'23]

- 基本一つのトークンが一つの音符を表し，各トークン（音符）は6つの変数の組で表現

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

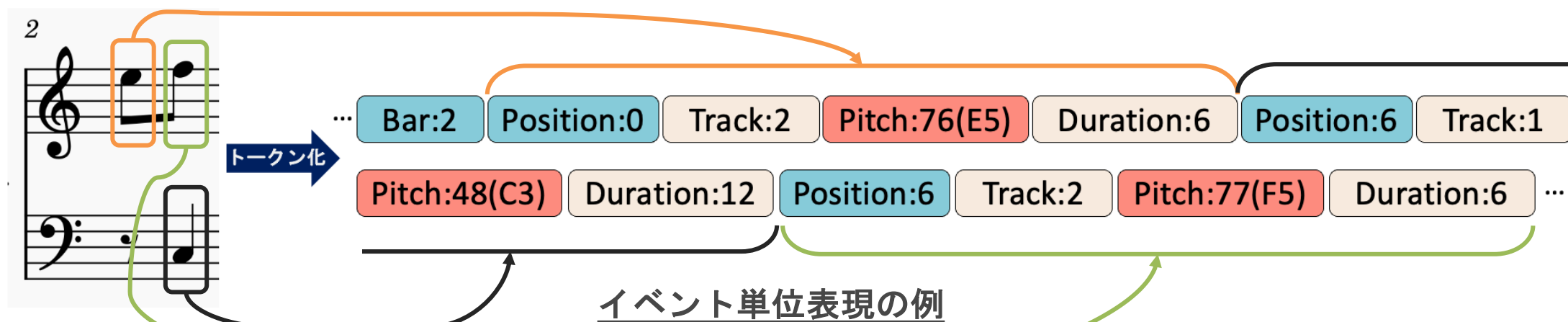
- エンコードの際には各変数を別々に線形変換層により変換して和を取る
 - デコードの際には出力の潜在表現に対し6種類の線形変換層により各変数を別々にデコード
- 各トークン種類ごとに辞書を持つ（語彙サイズは順に 3, 16, 48, 3, 128, 27）



音符単位表現の例

イベント単位表現 [Oore+'18; Huang+'20]

- 時刻と音高でソートした音符を順に Position, Track, Pitch, Duration の4トークンで表現し、これを順にトークン系列に追加
- 小節が変わった際には Bar トークンを追加
- トークン列の最初と最後にそれぞれ BOS トークンと EOS トークン
- 辞書は一つのみ（語彙サイズは $2+16+48+3+128+27=224$ ）



デコーダー型 Transformer による音楽生成

- 音符単位表現・イベント単位表現等を使うことで Transformer で音楽を処理できる
 - ・ 生成タスクにも理解タスクにも利用可能

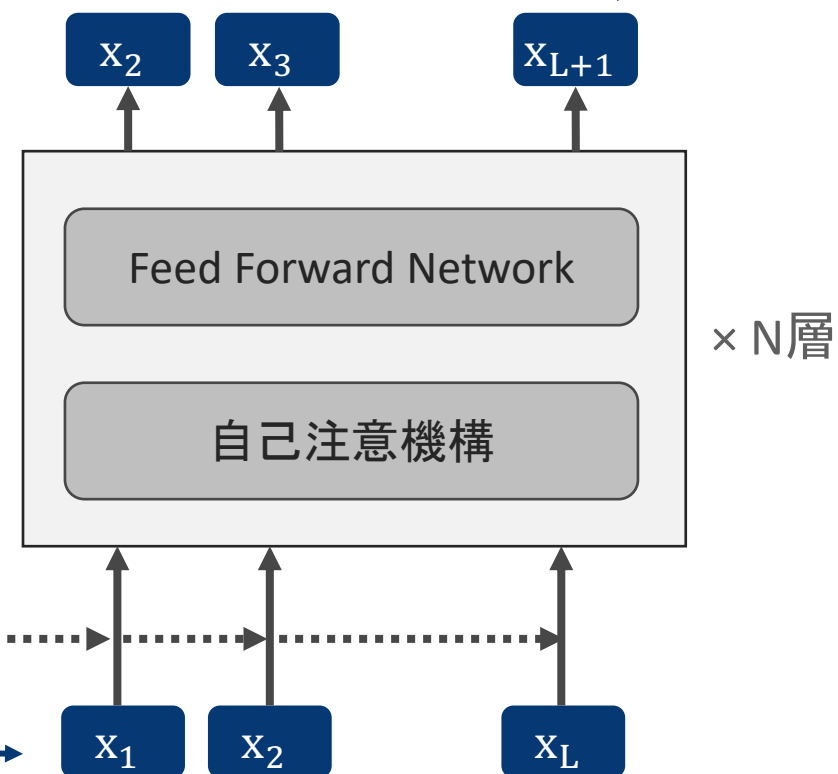


トークン化



音符単位表現
or
イベント単位表現

$\{X_1, X_2, \dots, X_L\}$



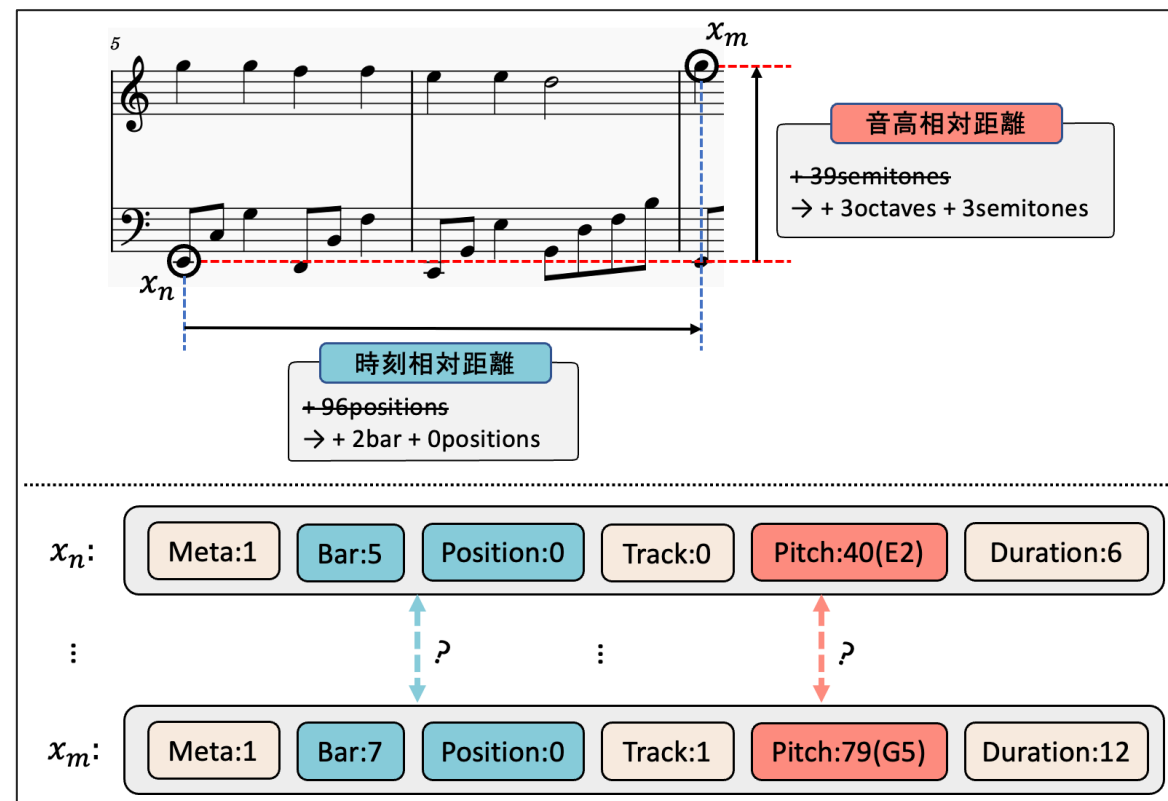
ところで

音楽における相対性と循環性

- 相対性: 音符間の時刻と音高の相対性は重要
 - 時刻: 時刻シフトした音楽は同じ音楽と認識可能
 - 音高: 移調された音楽も同じ音楽と認識可能
- 循環性: 各方向における循環性も音楽構造上重要
 - 時刻: 特定小節 (1,2,4,8) 間隔ごとに似たテーマが演奏される音楽は多い
 - 音高: 1オクターブ高い音楽は元の音の2倍の振動数を持ち, 似た響きを持つ

問題点: 相対性・循環性の欠落

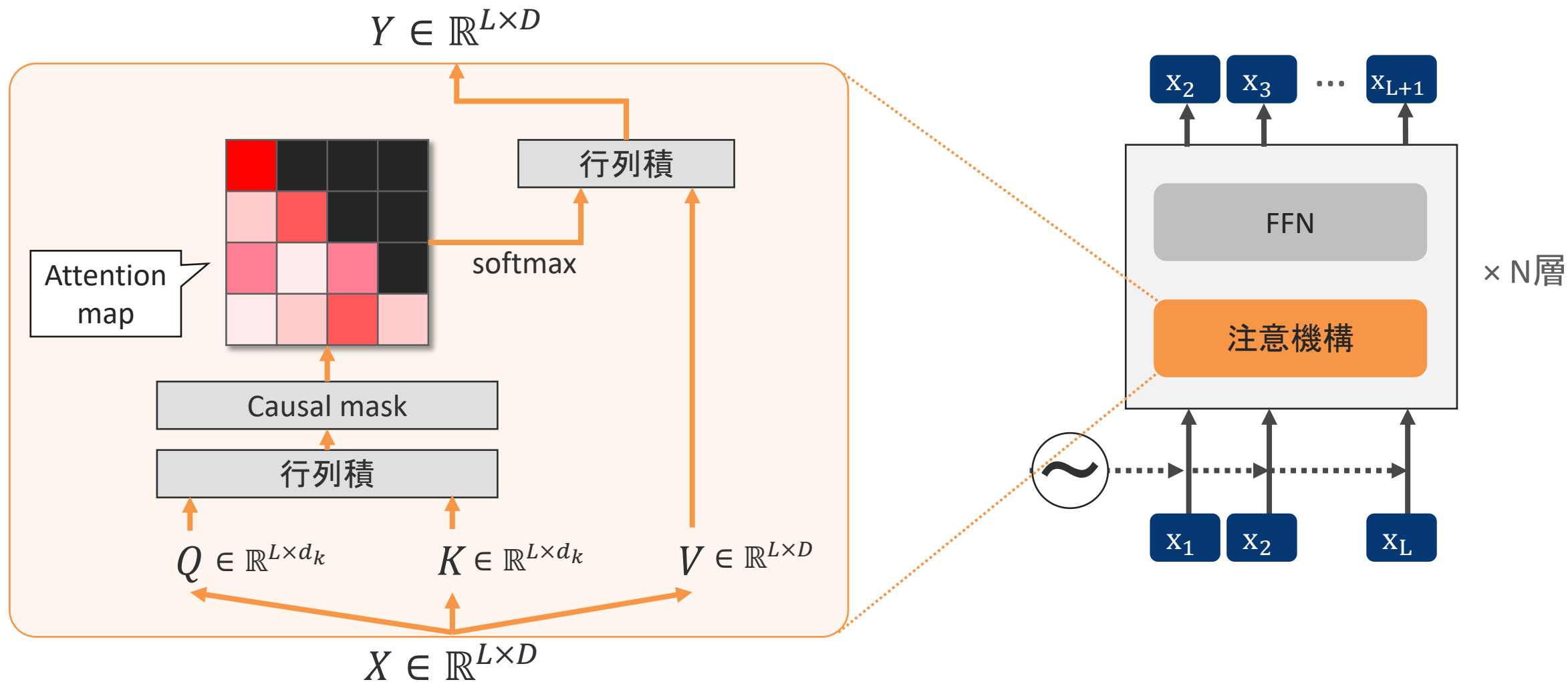
- 音符の絶対的時刻・音高の情報は含まれている一方、音符間の**相対距離**とその相対距離に内在する**循環性**が表現できていない
- 右図の音符単位表現の例において以下の情報は表現できていない
 - x_n と x_m の時刻相対距離が 96 positions (i.e., 2小節)
 - x_n と x_m の音高相対距離が 39 半音 (i.e., 3 オクターブと 3 半音)



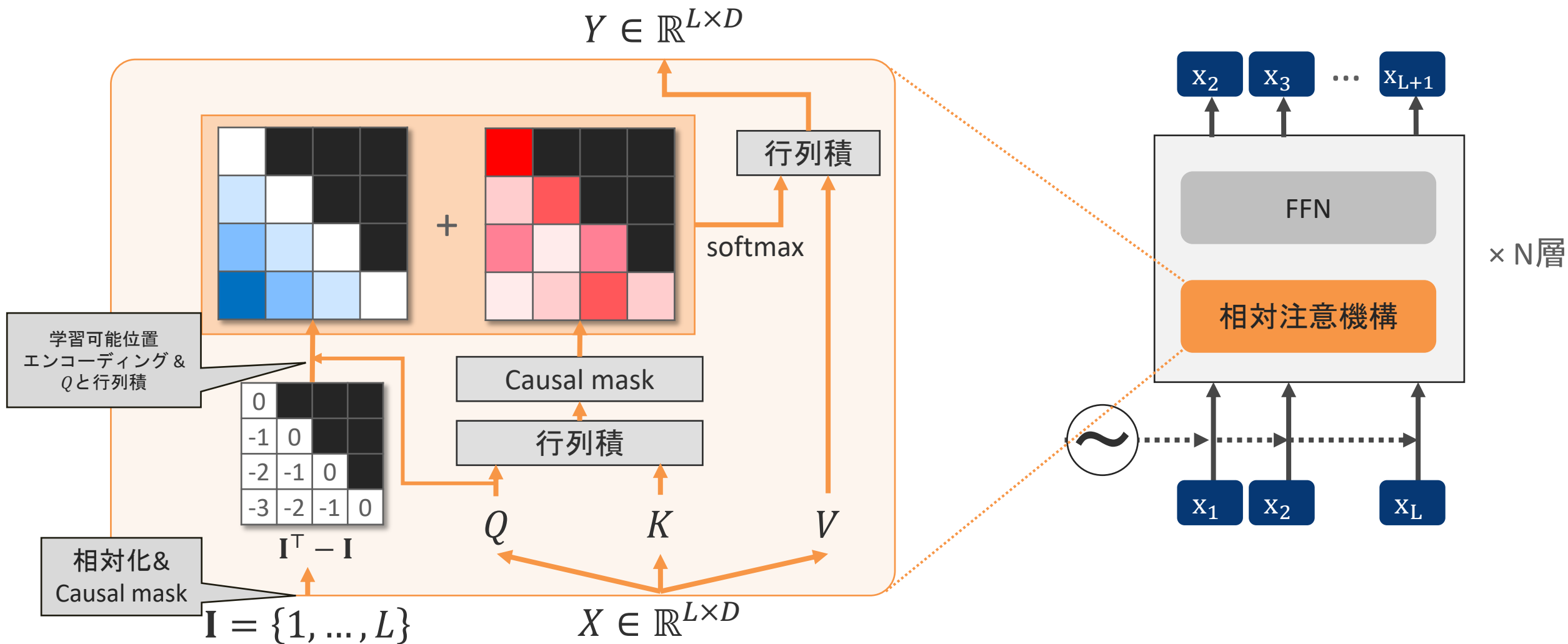
先行研究

- 自己注意機構を改良することで、トークン間の相対的な位置関係を効果的に捉えるアーキテクチャ
 - 相対注意機構 (Relative Attention): トークン間の相対インデックス距離を自己注意機構に取り込む
 - RIPO 注意機構 (RIPO Attention): トークン間の元の音楽上における相対時刻距離と相対音高距離を自己注意機構に取り込む
- 相対性はある程度補完できている一方、循環性については全く考慮されていない

注意機構




相対注意機構: 相対インデックス距離を取り込む



RIPO 注意機構: 元の音楽上における相対時刻距離と相対音高距離を取り込む

- 元の音楽における音符間の時刻と音高の相対距離を注意機構に取り込む
- 時刻系列 $\mathbf{T} = \left\{ x_i^{\text{bar}} \times \text{BarRes} + x_i^{\text{position}} \right\}_{i=1}^L$ と音高系列 $\mathbf{P} = \left\{ x_i^{\text{pitch}} \right\}_{i=1}^L$ を定義
 - 上記は音符単位表現における時刻系列と音高系列
 - 各トークン位置の時刻と音高をそれぞれ表している

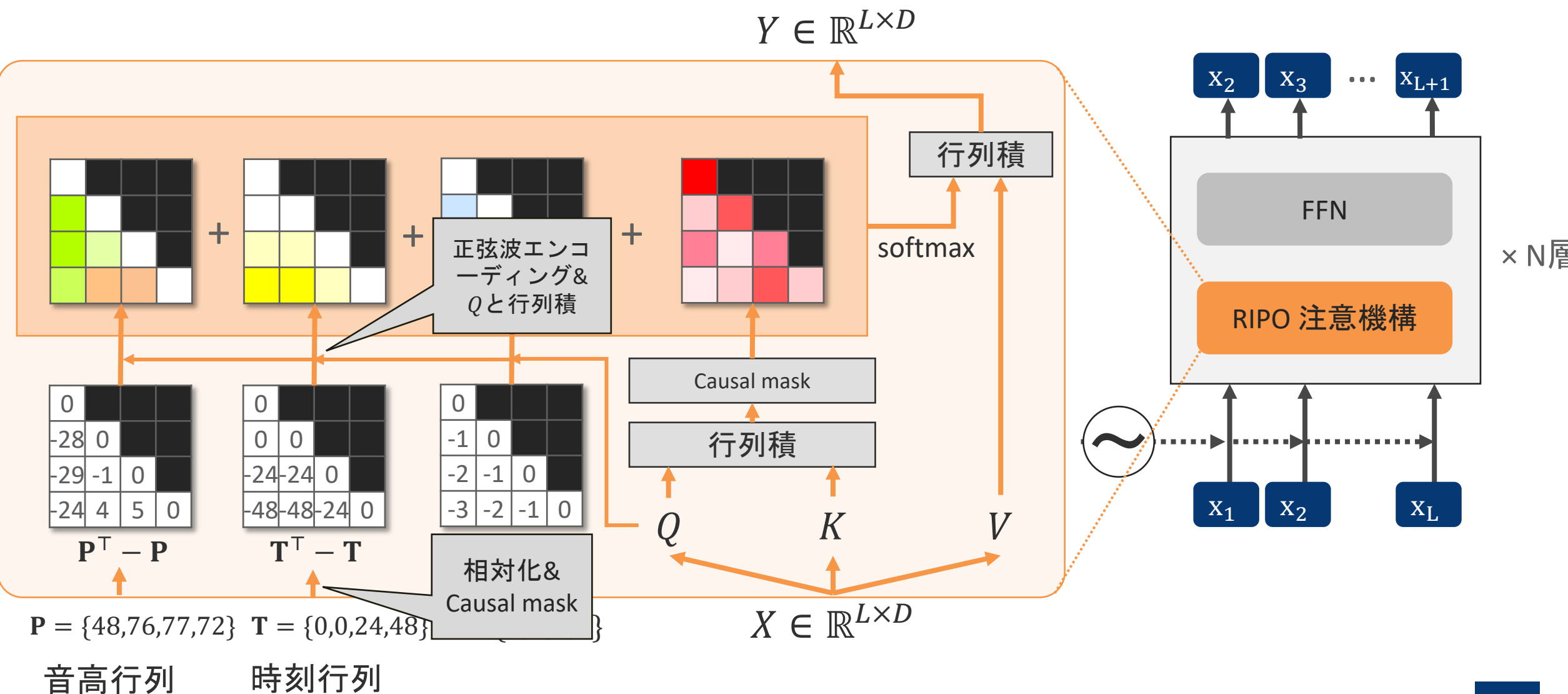
例:



$\mathbf{T} = \{0, 0, 24, 48\}$

$\mathbf{P} = \{48, 76, 77, 72\}$

RIPO 注意機構: 元の音楽上における相対時刻距離と相対音高距離を取り込む



先行研究

- 自己注意機構を改良することで、トークン間の相対的な位置関係を効果的に捉えるアーキテクチャ
 - 相対注意機構 (Relative Attention): トークン間の相対インデックス距離を自己注意機構に取り込む
 - RIPO 注意機構 (RIPO Attention): トークン間の元の音楽上における相対時刻距離と相対音高距離を自己注意機構に取り込む
- 相対性はある程度補完できている一方、循環性については全く考慮されていない

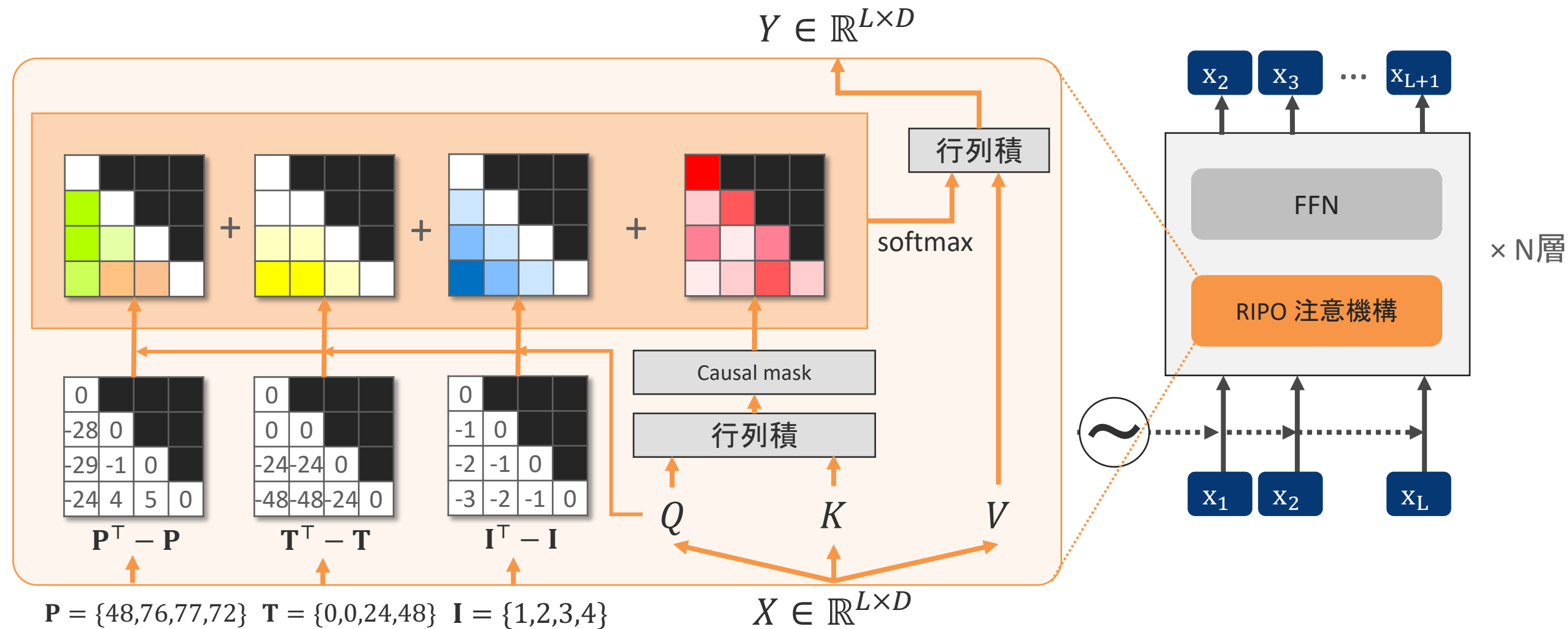
先行研究

- 自己注意機構を改良することで、トークン間の相対的な位置関係を効果的に捉えるアーキテクチャ
 - 相対注意機構 (Relative Attention): トークン間の相対インデックス距離を自己注意機構に取り込む
 - RIPO 注意機構 (RIPO Attention): トークン間の元の音楽上における相対時刻距離と相対音高距離を自己注意機構に取り込む
- 相対性はある程度補完できている一方、循環性については全く考慮されていない

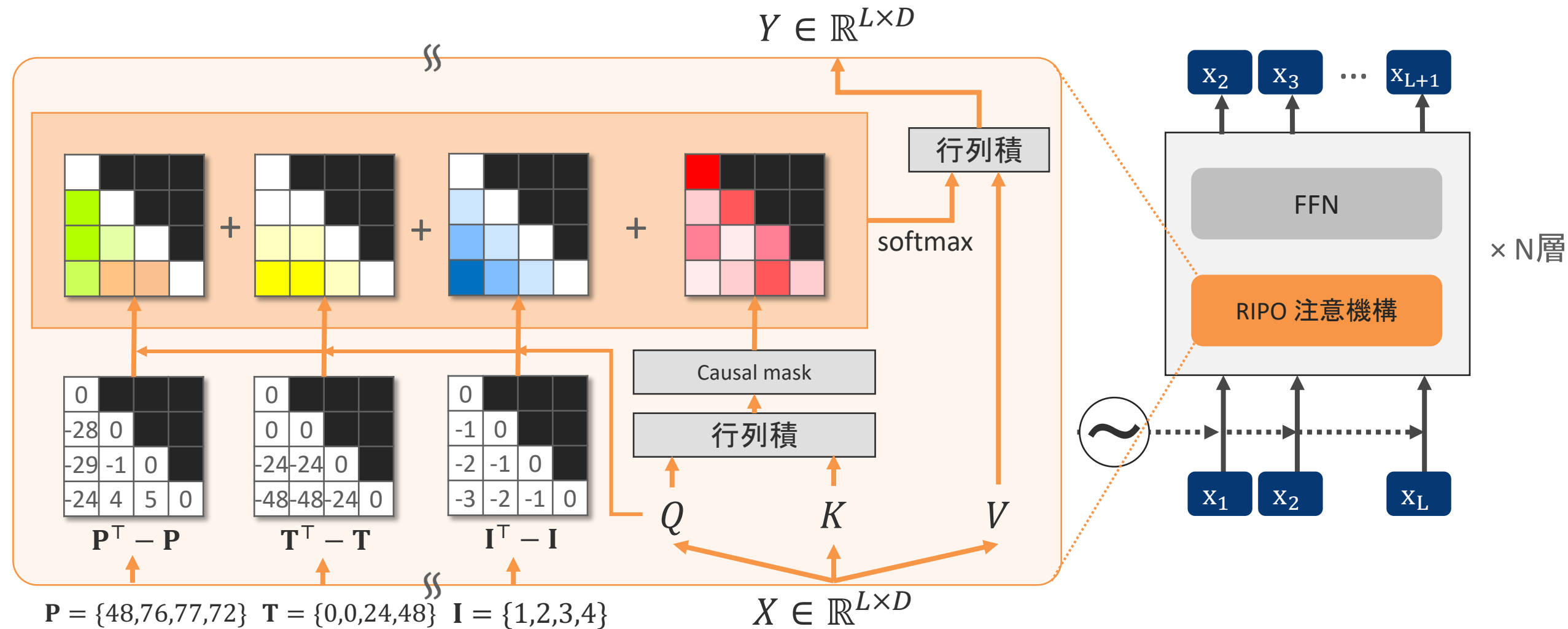
提案手法: 循環相対注意 (1/2)

- 相対性に加え循環性も考慮
- 時刻相対距離を小節単位とその余りに、音高相対距離をオクターブ単位とその余りに分解してそれぞれ別々で埋め込む
- 埋め込みには学習可能エンコーディングを使用
 - 距離それぞれに固有の関係性があるため
- 埋め込んだ後に時刻と音高それぞれでまとめてから注意機構に組み込む
 - 二種類のまとめ方
 - 要素ごとの足し算: 循環相対-S
 - 要素ごとの掛け算 (hadamard 積): 循環相対-H

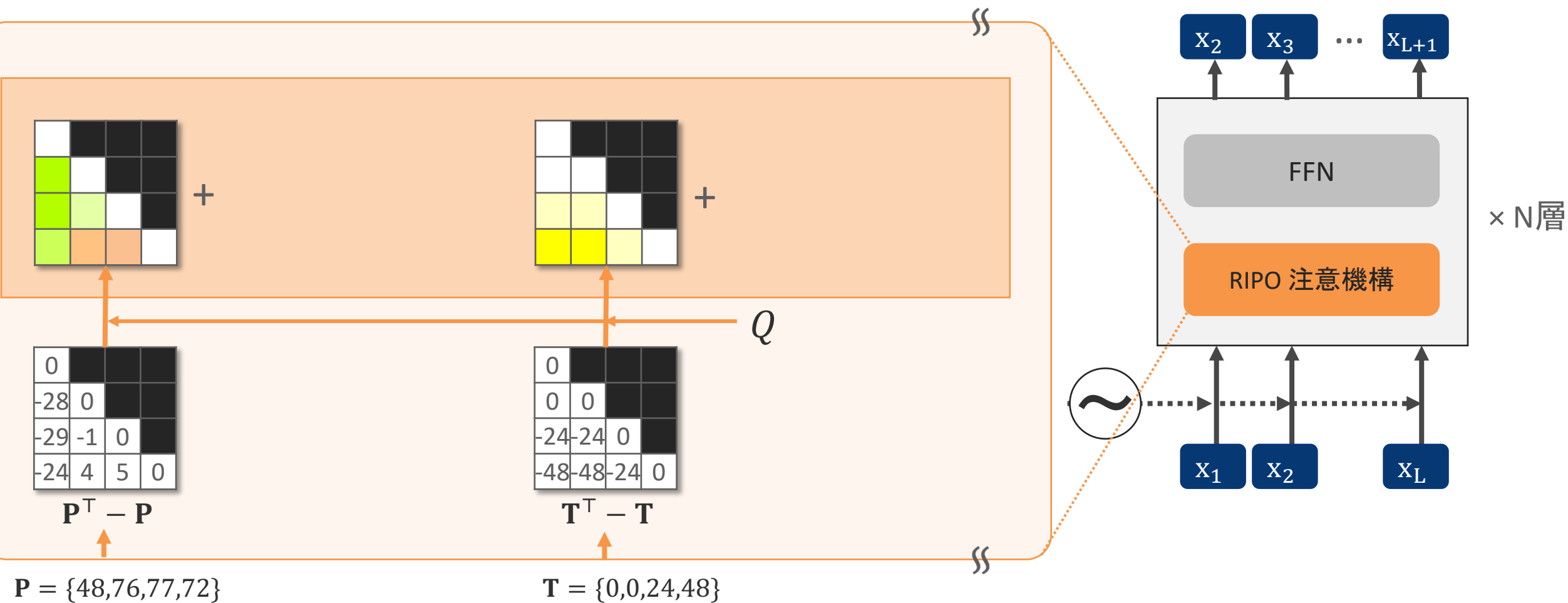
RIPO 注意機構



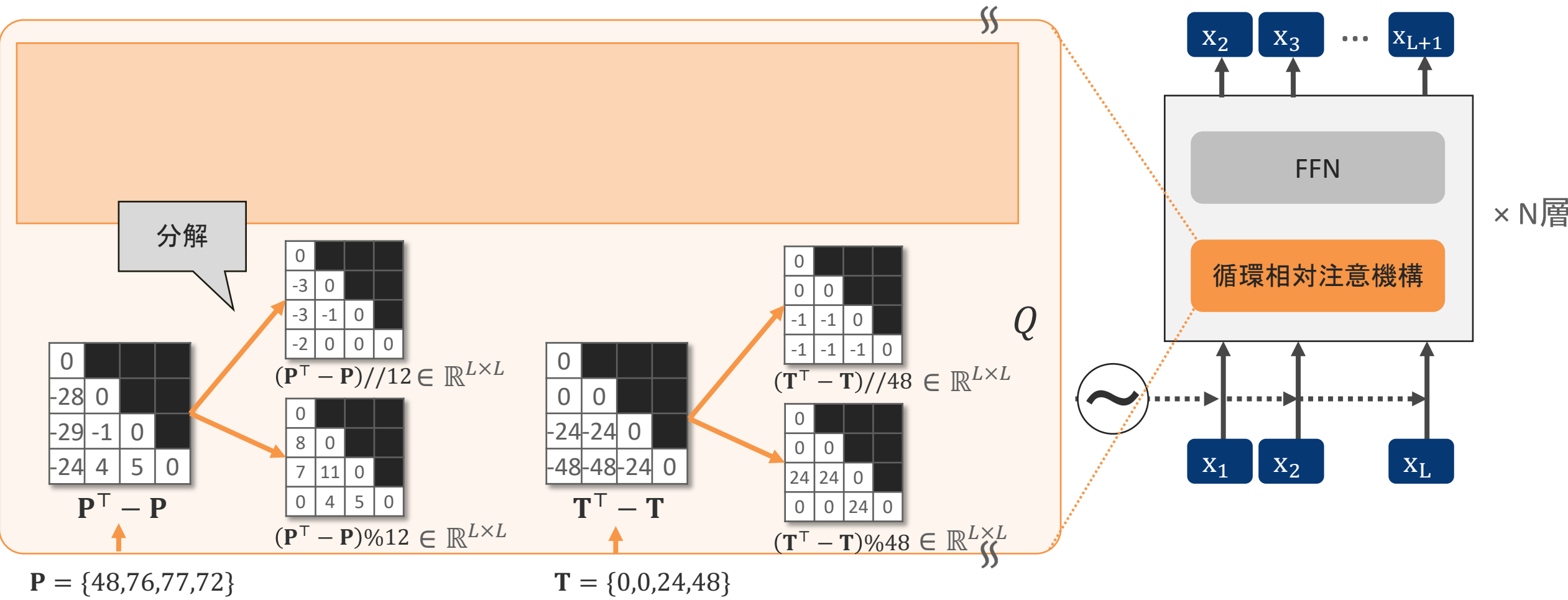
RIPO 注意機構



RIPO 注意機構

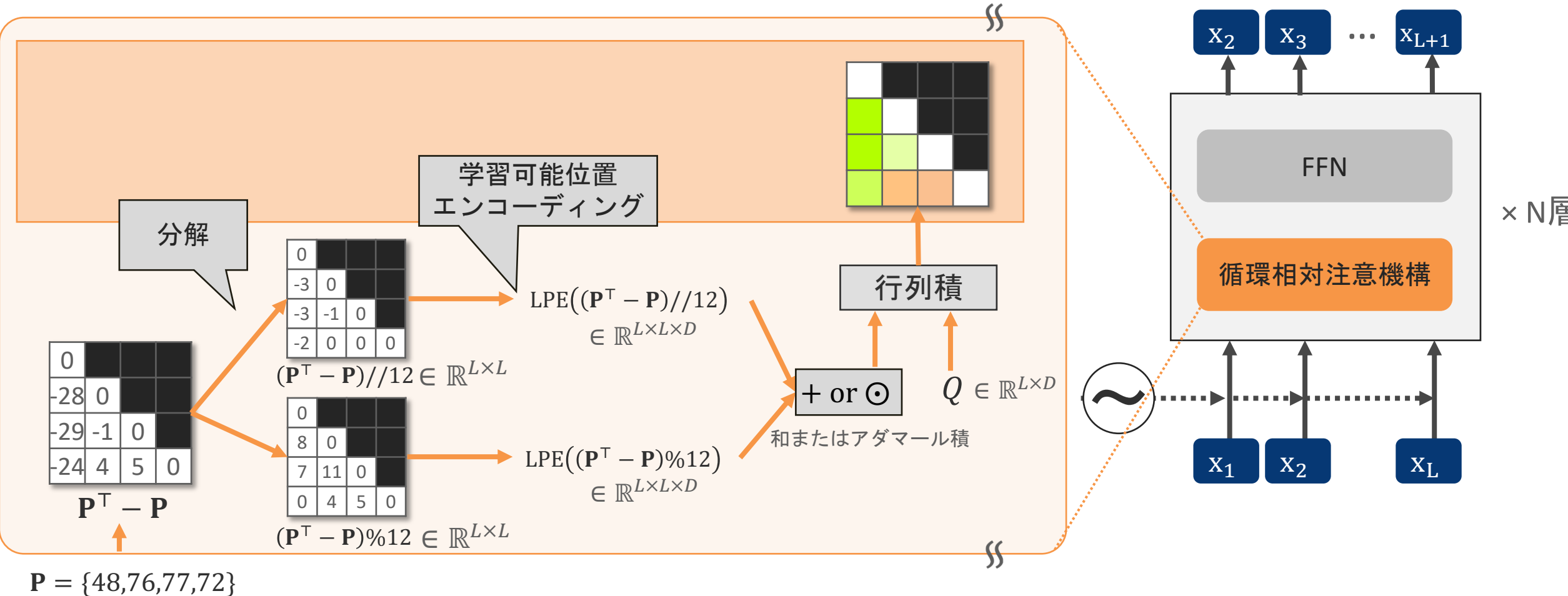


循環相対注意機構



音高は12(音高クラス数)を基に分解
時刻は48(一小節のステップ数)を基に分解

循環相対注意機構



音高は12(音高クラス数)を基に分解
時刻は48(一小節のステップ数)を基に分解

実験設定

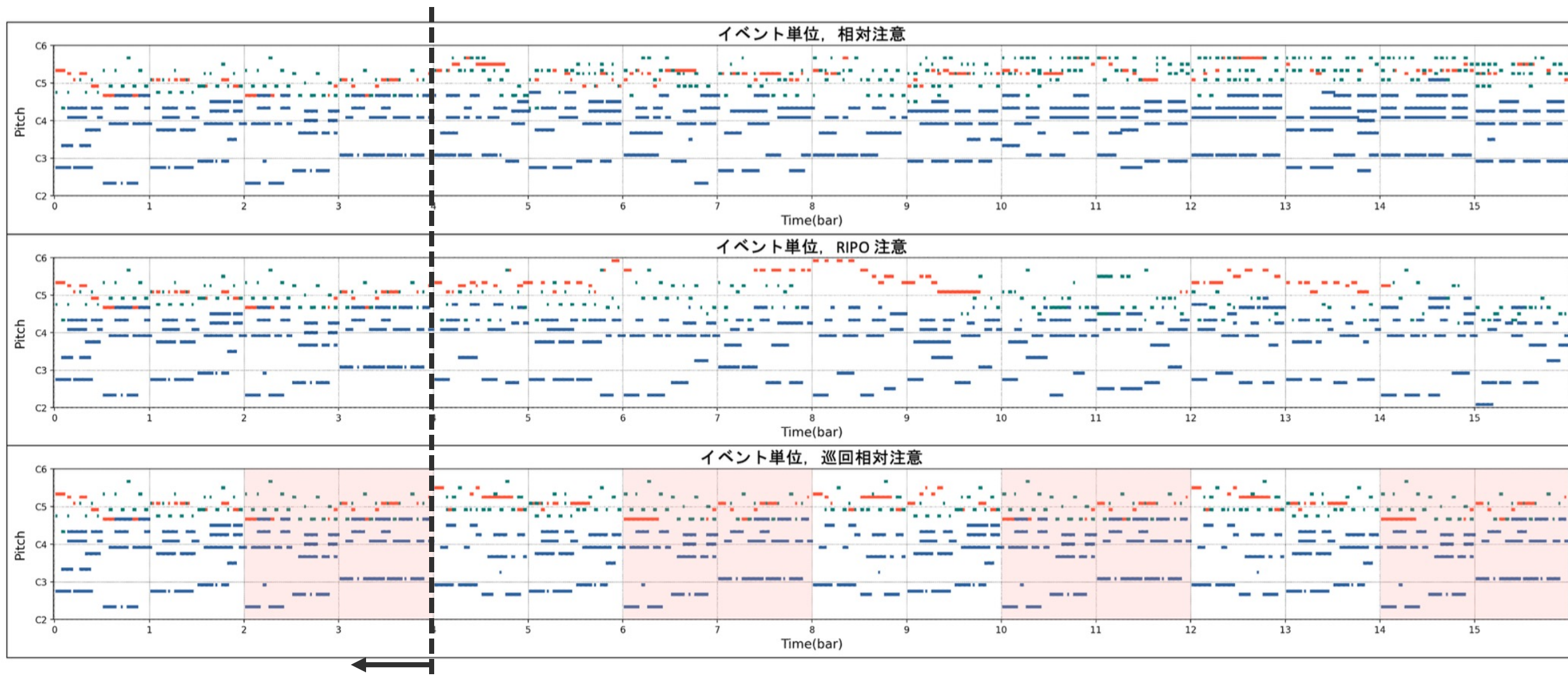
- モデル: 4層 の decoder 型 Transformer
- データセット: POP909
 - 繰り返し構造が豊富に含まれるポピュラー音楽のデータセット
 - 16小節となるようにストライド1小節でデータ作成
 - 学習データ: 35,452件, 検証データ: 4,749件, テストデータ: 4,137件
- ベースライン: 注意機構・相対注意機構・RIPO 注意機構

実験: 主観評価 (1/2)

- 後続生成タスク
 - 4小節をモデルに与え, 続きの12小節を生成させる
 - 各サンプルセットごとに3人が評価
 - 10のサンプルセットで評価
- 評価項目(それぞれ1~5の5段階評価)
 - 一貫性・音楽性・総合点

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

実験: 主観評価 (2/2)



プロンプト



実験: 客観評価

- 15小節を与え続きの1小節を生成させ、それがテストデータとどれだけ似ているか
- 循環相対-H がほとんどの指標において既存手法を上回るスコア

手法						客観評価					
No.	表現方法	注意機構	Params	Time (ms/note)	損失	$F1_{note}$	$F1_{pr}$	ND	CS	GS	PRS
1	イベント単位	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		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	音符単位	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		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

音符単位でどれだけ一致しているか

ピアノロール上でどれだけ一致しているか

音符密度

クロマベクトルの類似度

リズムの類似度

音高範囲の類似度

まとめ

- 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
- 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
- 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