

GPT-4におけるメロディ生成能力の検証

刀柝 賢志^{1,a)} 北原 鉄朗^{1,b)}

概要: 本研究では、GPT-4 に対して音楽理論を用いたプロンプトを与えてメロディ生成が可能かどうかを検討する。GPT-4 を用いたメロディ生成は一部において試行されているものの、音楽理論で用いられる様々な用語をプロンプトとして利用できるかについては、十分な検証は行われていない。本発表では、ジャズにおけるメロディ生成に着目し、スケール、コード進行、コードスケールを表すプロンプトを GPT-4 に与え、当該音楽理論を守ったメロディが生成されるかどうかを評価した。コード進行、スケールの構成音を明示した場合には十分な結果を得る事ができたが、構成音を間接的に指示した場合、十分な結果は得られなかった。また、コードスケールに関しても、コードから正しいスケールを選び、指示通りの結果を得る事はできなかった。

1. はじめに

作曲は、知的な創作行為であるため、誰でも簡単にできるものではない。そのため、たとえば自作の歌詞にメロディをつけたいというような場面があったとしても、実際にメロディを作成するには、音楽理論を学び、メロディ作成のノウハウを身に着ける必要がある。そのことから、音楽は多くの人が触れている娯楽であるにもかかわらず、作曲という形で音楽を楽しむ人は、ごく少数に限られる。

手軽に自作作曲を入手できるようにするための研究として、自動作曲や作曲支援の研究が行われている [1], [2], [3], [4], [5], [6]。Yu は、歌詞からメロディを生成するための LSTM と GAN を組み合わせた深層生成モデルを提案した [7]。Zhang [3] は、歌詞とメロディの調和を確保するために音楽理論に基づく歌詞とメロディの関係を組み込んだ手法「ReLyMe」を提案した [8]。

自動作曲や作曲支援における 1 つの課題が、ユーザがほしいメロディの特徴をシステムやモデルにどのように伝えるかである。これに対する 1 つのアプローチに、近年急速に発展が続いている大規模言語モデル (LLM) の利用が考えられる。つまり、メロディの特徴をテキストとして与え、LLM がそれを解釈してメロディに変換するアプローチである。実際、Yuan らは、音楽能力を統合した LLM 「ChatMusician」を提案した [9]。ABC 表記法というテキスト互換型音楽表現を用いて LLM の学習を行い、テキスト同士の変換の形で音楽生成を実現した。川口らは、ChatGPT

のプロンプトエンジニアリングの枠組みを利用し、対話的な作曲作業のフレームワークのデザインに取り組んでいる [10]。

音楽には、コード進行、音階 (スケール)、音価などに関する様々な専門用語が存在する。音楽家たちは、これらの用語を正しく理解し使用することで、他の音楽家と協調的に作曲作業を行うことができる。そのため、LLM がこれらの用語を正しく理解する能力を有するかどうかは、検証すべき重要な課題である。

本稿では、代表的な LLM の 1 つである GPT-4 を用いて、メロディ生成における条件に主要な音楽用語を指定したときに、その条件を守ったメロディを生成するかどうかを検証する。たとえば、「ハ長調のスケールの構成音のみでメロディ生成を行ってください」のようなプロンプトを与えたときに、実際に、生成されたメロディがハ長調のスケールの構成音のみで構成されているかどうかを評価する。コード進行、スケールのそれぞれについて、このような実験を行う。

2. 実験方法

2.1 実験の概要

本稿では、GPT-4 に対して、メロディに関する制約を与えてメロディを生成させるプロンプトを与えたときに、その制約をどの程度守ったメロディが出力されるか検証する。次の 3 つの実験により構成する。

実験 1 コードの構成音を明示し、当該構成音のみでメロディを生成

実験 2 メロディで使うスケールを指定

¹ 日本大学文理学部情報科学科

^{a)} tone@kthrlab.jp

^{b)} kitahara@kthrlab.jp

実験 3 コード進行とスケールの対応関係を明示し、使うスケールを自動で選んでメロディを生成
各実験は、さらに条件を複数設定する。同じ条件に対して100回実験を繰り返し、制約を守った音符数の割合を議論する。

2.2 共通のプロンプト

本稿の実験では、コードが1小節ごとに切り替わること、4/4拍子であること、メロディは単旋律であることを前提とする。また、出力形式にはABC Music Notationを用いる。そこで、次のプロンプトを全実験の共通プロンプトとして使用する。

共通プロンプト

あなたは音楽家です。以下の制約に従ってメロディを作曲してください。

制約：コードは1小節で切り替わります

制約：ABC Music Notation形式で表示してください

制約：4/4拍子で作成してください。

制約：できたメロディ以外は絶対表示しないでください

制約：日本語を表示しないでください

制約：メロディは常に1つの音のみで構成し、音を重ねないでください

制約：コード進行は絶対に表示しないでください

制約：メロディの単旋律のみを表示してください

2.3 【実験 1】コード構成音の指定

コード進行を指定し、各コードの構成音を指示した上で、そのコード構成音のみでメロディ生成を行うように指示する。コード進行は、Em7 - A7 - Dm7 -G7 - Cmaj7 とする。コードの構成音の指示は、次の3種類を試行する。

方針 1 コード構成音の音名を直接表記

方針 2 ルート音を示し、隣り合う構成音同士の音程を「+3」などと表記

方針 3 ルート音を示し、各構成音のルート音からの音程を「+3」などと表記

使用するプロンプト

コード進行そのものを指定するプロンプトは、方針1~4に共通であり、次の通りである。

コード進行を指定するプロンプト

制約：コード進行はEm7 - A7 - Dm7 -G7 - Cmaj7とします

制約：コード進行は絶対に変更しないでください

制約：コードは1小節で切り替わります

制約：各小節、その小節のコードの構成音のみでメロディ生成をしてください

制約：各小節、その小節のコード構成音以外は絶対に使用しないでください

各コードの構成音を指示するプロンプトのうち、方針1は次のものを用いる。

コード進行を指定するプロンプト

制約：以下に各コードの構成音を明記します

Em7: E, G, B, D

A7: A, ^C, E, G

Dm7: D, F, A, C

G7: G, B, D, F

Cmaj7: C, E, G, B

各小節、明記したコードの構成音のみでメロディを生成してください

方針2, 3では、「半音」という概念をGPT-4を理解する必要がある。そこで次のプロンプトを与える。

全音・半音を教えるプロンプト

制約：以下に、全音と半音についての説明を記述します

半音は鍵盤上で隣り合っている2つの音の距離のことを指します

半音では2つの音の間に音は存在しません

半音の例) cと.c, dと^d, bとc

全音は2つの音の間に、1つ音を挟んでいる場合の距離を指します

つまり全音は、半音2つ分の距離です

全音の例) cとd, gとa, bと^c

このプロンプトののちに、次のプロンプトを与える。

方針 2 におけるコード構成音の指示

制約：以下に各コードの構成音について説明します

一番低い音(ルート音)を明示し、1つ前との音の差を明示します

Em7: E,+3,+4,+3

A7: A,+4,+3,+3

Dm7: D,+3,+4,+3

G7: G,+4,+3,+3

Cmaj7: C,+4,+3,+3

この構成音だけでメロディ生成を行ってください

方針 3 におけるコード構成音の指示

制約：以下に各コードの構成音について説明します

一番低い音(ルート音)を明示し、1つ前との音の差を明示します

Em7: E,+3,+7,+10

A7: A,+4,+7,+10

Dm7: D,+3,+4,+3

G7: G,+4,+3,+3

Cmaj7: C,+4,+7,+10

この構成音だけでメロディ生成を行ってください

2.4 【実験 2】スケールの指定

スケールの構成音を指示し、その構成音のみを使ってメロディを作るように指示する。使用するスケールは次の9個である。

- S1: ハ長調

- S2：ハ短調
- S3：C Harmonic Minor
- S4：C Natural Minor
- S5：C Melodic Minor
- S6：C Whole Tone
- S7：C Combination of Diminished
- S8：C Major pentatonic
- S9：C Minor Pentatonic

各スケールのルート音はCとする。「ハ短調」は実質的には「Natural Minor Scale」と等しいことが予想されるが、表記による違いを見るため、両方を実験対象とする。各スケールの構成音は、実験1の方針1~3と同じ3つの方針で指示する。

使用するプロンプト

スケール名の指定は、方針1~3にかかわらず次のプロンプトを共通で用いる。

スケール実験で共通のプロンプト

制約：スケールは {使用するスケール名} とします
 制約：スケールの構成音のみでメロディ生成を行ってください
 制約：スケールの構成音以外の音は絶対に使用しないでください

スケールの構成音を直接指定する方針1では、次のプロンプトを用いる。

方針1におけるスケール構成音の指示

条件：ハ長調のスケール構成は以下の通りです
 C, D, E, F, G, A, B
 このスケール構成音でメロディを生成してください

方針2,3では、実験1と同様、半音と全音の概念をGPT-4に理解させる必要がある。次のプロンプトを用いる。

全音・半音を教えるプロンプト

制約：以下に、全音と半音についての説明を記述します
 半音は鍵盤上で隣り合っている2つの音の距離のことを指します
 半音では2つの音の間に音は存在しません
 半音の例) cと_c, dと^d, bとc
 全音は2つの音の間に、1つ音を挟んでいる場合の距離を指します
 つまり全音は、半音2つ分の距離です
 全音の例) cとd, gとa, bと^c

その上で、各スケールの構成音について次のプロンプトで指示する。

方針2におけるスケール構成音指示

制約：以下にハ長調の構成音について説明します
 一番低い音（ルート音）を明示し、1つ前との音の差を明示します
 (ルート音) C, +2, +2, +1, +2, +2, +2, +1
 この構成音だけでメロディ生成を行ってください

方針3におけるスケール構成音指示

制約：以下にハ長調の構成音について説明します
 一番低い音（ルート音）を明示し、1つ前との音の差を明示します
 (ルート音) C, +2, +2, +1, +2, +2, +2, +1
 この構成音だけでメロディ生成を行ってください

2.5 【実験3】コード進行とスケールの関係を指示

いくつかのコード進行に対して、そのコード進行に合うスケールを教えた上で、コード進行のみを指示したときに、それに対応するスケールの構成音を使ったメロディを生成するか確かめる。ここで取り上げるコード進行およびスケールは、次の通りである。

- P1：C, Dm, Em, F, G → Cmaj
- P2：Am, Bm, C, Dm, Em → A Natural Minor
- P3：Am, Bm, C, Dm, E → A Harmonic Minor
- P4：Am, Bm, C, D, E → A Melodic Minor

(左：コード進行, 右：スケール名)

使用するプロンプト

次のように、コード進行とスケールの関係を指示した後、スケールの構成音のみでメロディを作成するよう指示する。

コードスケール実験

制約：以下にコード進行に対するスケールの対応を評価する。
 これは、コード進行が与えられた時に、キーを定めるものである。
 1, (コード進行) C, Dm, Em, F, G (キー) Cmaj
 2, (コード進行) Am, Bm, C, Dm, Em (キー) A Natural Minor
 3, (コード進行) Am, Bm, C, Dm, E (キー) A Harmonic Minor
 4, (コード進行) Am, Bm, C, D, E (キー) A Melodic Minor
 以上がコード進行とスケールの対応です。
 この中で定めたキーの構成音だけでメロディを生成してください
 制約：以下に、各スケールの構成音を示します。
 Cmaj:C, D, E, F, G, A, B
 A Natural Minor:C, D, _E, F, G, _A, _B, C
 A Harmonic Minor:C, D, _E, F, G, _A, B, C
 A Melodic Minor:C, D, _E, F, G, A, B, C
 制約：コード進行は (指定するコード進行) とします。
 制約：選択したスケールの構成音のみでメロディを生成してください

3. 実験結果

各実験の各条件において、メロディの生成を100回行い、試行ごとに与えた制約を満たす音符の割合を求めた。その割合が、100%, 90%以上 100%未満, 80%以上 90%未満, 70%以上 80%未満, 60%以上 70%未満, 50%以上 60%未満, 50%未満のものの試行数を求めた。また、エラーになる場合

表 1 実験 1 (コード構成音の指定) の結果

| | 100% | 90% | 80% | 70% | 60% | 50% | <50% | エラー |
|------|------|-----|-----|-----|-----|-----|------|-----|
| 方針 1 | 66 | 17 | 2 | 2 | 2 | 1 | 1 | 9 |
| 方針 2 | 20 | 51 | 9 | 1 | 0 | 0 | 0 | 19 |
| 方針 3 | 12 | 52 | 11 | 2 | 3 | 2 | 3 | 15 |

表 2 実験 2 (スケール構成音の指示) 「方針 1」 の結果

| | 100% | 90% | 80% | 70% | 60% | 50% | <50% | エラー |
|----|------|-----|-----|-----|-----|-----|------|-----|
| S1 | 90 | 0 | 0 | 0 | 0 | 0 | 0 | 10 |
| S2 | 88 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| S3 | 1 | 16 | 76 | 2 | 0 | 0 | 0 | 5 |
| S4 | 97 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| S5 | 1 | 3 | 17 | 55 | 20 | 1 | 0 | 8 |
| S6 | 83 | 5 | 0 | 1 | 1 | 3 | 1 | 6 |
| S7 | 87 | 3 | 1 | 0 | 0 | 0 | 0 | 9 |
| S8 | 94 | 2 | 0 | 0 | 0 | 0 | 0 | 4 |
| S9 | 26 | 0 | 2 | 0 | 7 | 2 | 0 | 63 |

表 3 実験 2 (スケール構成音の指示) 「方針 2」 の結果

| | 100% | 90% | 80% | 70% | 60% | 50% | <50% | エラー |
|----|------|-----|-----|-----|-----|-----|------|-----|
| S1 | 85 | 0 | 0 | 0 | 0 | 0 | 0 | 15 |
| S2 | 50 | 0 | 1 | 0 | 0 | 0 | 1 | 48 |
| S3 | 9 | 17 | 40 | 7 | 2 | 0 | 0 | 25 |
| S4 | 88 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| S5 | 0 | 2 | 19 | 46 | 20 | 3 | 1 | 9 |
| S6 | 0 | 0 | 0 | 9 | 15 | 41 | 17 | 18 |
| S7 | 1 | 1 | 5 | 16 | 30 | 17 | 1 | 29 |
| S8 | 72 | 11 | 2 | 0 | 0 | 0 | 0 | 15 |
| S9 | 4 | 2 | 27 | 25 | 14 | 2 | 0 | 26 |

の試行数も求めた。ここでいうエラーとは、GPT-4 の出力が ABC Notation の仕様を満たさない場合などである。以下、実験ごとに結果を述べる。

3.1 【実験 1】コード構成音の指示

実験 1 の結果を表 1 に示す。コードの構成音を直接指示する「方針 1」では、100 回中 83 回においてメロディの 90% 以上の音符がコード構成音からなっていたのに対し、ルート音およびそこから音程というように間接的に指示する「方針 2」「方針 3」では、制約を満たす音符の割合が大幅に低下した。

3.2 【実験 2】スケール構成音の指示

実験 2 の結果を表 2、表 3、表 4 に示す。スケールの構成音を直接指示する「方針 1」では、各スケールにおいて、メロディの 89% 以上の音符がコード構成音からなっていたのに対し、主音からの音程の形で間接的に指示する「方針 2」「方針 3」では、実験 1 と同様に、制約を満たす音符の割合が大幅に低下した。また、「方針 1」の S9 では、約 3 割がエラーとなった。

3.3 【実験 3】コード進行とスケールの関係を指示

実験 3 の結果を表 5 に示す。全ての実験で、半数以上

表 4 実験 2 (スケール構成音の指示) 「方針 3」 の結果

| | 100% | 90% | 80% | 70% | 60% | 50% | <50% | エラー |
|----|------|-----|-----|-----|-----|-----|------|-----|
| S1 | 81 | 0 | 0 | 0 | 0 | 0 | 0 | 19 |
| S2 | 43 | 1 | 0 | 1 | 0 | 0 | 0 | 55 |
| S3 | 9 | 29 | 36 | 3 | 2 | 0 | 0 | 21 |
| S4 | 76 | 0 | 0 | 0 | 0 | 0 | 0 | 24 |
| S5 | 11 | 11 | 26 | 23 | 13 | 2 | 0 | 24 |
| S6 | 0 | 0 | 0 | 2 | 9 | 56 | 24 | 21 |
| S7 | 2 | 3 | 13 | 19 | 28 | 5 | 0 | 30 |
| S8 | 71 | 5 | 2 | 0 | 0 | 0 | 0 | 22 |
| S9 | 12 | 5 | 34 | 24 | 9 | 3 | 0 | 13 |

表 5 実験 3 (コード進行とスケールの関係を指示) の結果

| | 100% | 90% | 80% | 70% | 60% | 50% | <50% | エラー |
|----|------|-----|-----|-----|-----|-----|------|-----|
| P1 | 31 | 0 | 1 | 0 | 0 | 0 | 0 | 68 |
| P2 | 0 | 1 | 1 | 3 | 6 | 7 | 29 | 53 |
| P3 | 1 | 2 | 1 | 2 | 15 | 13 | 10 | 56 |
| P4 | 1 | 1 | 9 | 13 | 15 | 4 | 2 | 55 |

がエラーとなった。また、制約を満たす音符の割合として 100% を多く示したのは P1 のみで、他の実験では十分な正答率は得られなかった。

4. 考察

ここでは、音高に関する考察を行う。以下に、音高に関する考察結果を示す。

- 使用することのできる音高を明示した場合、高確率で指示に従う
 - C Major Scale の構成音以外の音の生成に弱い
 - スケールは、C Major Scale と C Natural Minor Scale の 2 種類しかない
 - コードからスケールを決定することはできない
- それぞれの考察をした理由について以下に記述する。

4.1 考察 1

音高を具体的に示す条件は、実験 1、実験 2 の「方針 1」が該当する。他の条件と比べてときに、正答率 100% が最頻値になるものは、具体的な音高を示した「方針 1」のみである。また、正答率 80% 以上の割合も他の実験に比べ高いことがわかる。以上のことから、具体的な音高を示した場合には、使うべき音高の指示に従うことができると言える。

4.2 考察 2

黒鍵部分の生成に弱いことについて考察する。例えば、実験 1 で扱ったコード進行は Em7 - A7 - Dm7 - G7 - Cmaj7 であるため、2 小節目に C# を使用しなくてはならない。一方、実際に生成されたメロディ (図 1) では、C# であるべき箇所でも C の音を生成している。

次に、実験 1 で生成されたメロディの例を図 2、図 3 に示す。いずれも、C Major Scale の構成音で構成されるメロディとなっており、指示されたスケールは使われていない。



図 1 実験 1 におけるメロディ生成例



図 2 実験 2 における C Natural Minor Scale のメロディ生成例



図 3 実験 2 における C Whole Tone Scale のメロディ生成例



図 4 実験 2 におけるハ長調のメロディ生成例



図 5 実験 2 における C Pentatonic Scale のメロディ生成例

以上から、C Major Scale 以外を使ったメロディの生成は難しいことと言える。

4.3 考察 3

ここでは、GPT-4 が Major と Minor の 2 種類のスケールしか使用しないことを示す。

長調に合うスケールとして Major Scale しか用いないことを示すため、C Pentatonic Scale の生成結果を述べる。ハ長調の構成音でのメロディ生成を指示した場合に生成されたメロディを図 4 に示す。この条件では、100%の正答率でハ長調のメロディを生成した。一方、C Pentatonic Scale を用いたときのメロディ生成結果を図 5 に示す。C Pentatonic Scale の校正音は C, D, E, G, A であるが、B が使用されており、Major Scale が使われていることがわかる。

短調に合うスケールとしては、プロンプトの指示に関わらず Natural Minor Scale を用いてメロディを生成していると考えられる。C Natural Minor Scale, C Harmonic Minor Scale, C Melodic Minor Scale を指示して生成されたメロディを図 6, 図 7, 表 8 に示す。3つのメロディはすべて C Natural Minor Scale の構成音で構成されている。実験結果では、C Natural Minor Scale では、正答率 100% のメロディが多く生成されているのに対し、C Harmonic Minor Scale, C Melodic Minor Scale では、正答率 70%, 80% のメロディが多く生成されていることから、このことが示唆される。



図 6 実験 2 における C Natural Minor Scale のメロディ生成例



図 7 実験 2 における C Harmonic Minor Scale のメロディ生成例



図 8 実験 2 における C Melodic Minor Scale のメロディ生成例



図 9 実験 3 におけるコード進行 P1 に対するメロディ生成例



図 10 実験 3 におけるコード進行 P2 に対するメロディ生成例



図 11 実験 3 におけるコード進行 P3 に対するメロディ生成例



図 12 実験 3 におけるコード進行 P4 に対するメロディ生成例

4.4 考察 4

実験 3 から、コード進行とスケールの関係性を教え、コード進行を指示することで適切なスケールを自動で選んでメロディを生成することは、難しいと考えられる。

コード進行が P1 の場合、正答率 100% を多く示したが、約 6割がエラーとなり、メロディを生成することはできてなかった。生成されたメロディの一例を表 9 に示す。

コード進行が P2, P3, P4 の場合で、生成されたメロディの正答率に大きな差は見られない。実際に生成されたメロディの例を表 10, 表 11, 表 12 に示す。いずれのメロディも C Major Scale で構成されていることから、プロンプト内の制約に従っていないことがわかる。また、約半数がエラーになっていることから、GPT-4 は、コードとスケールの関係性を理解することができないと言える。

5. まとめ

本稿では、GPT-4 を用いたメロディ生成能力について、

いくつかの観点から実験した。本実験を通して、GPT-4 におけるメロディ生成能力が見えてきた。コードとスケールの関係のような音楽的な知識を指示した後に、適切な音を選択させる実験では、指示した制約を満たすメロディを得ることができなかった。特に、Major Scale と Natural Minor Scale 以外を指示しても、Major Scale または Natural Minor Scale によるメロディが生成される事態が頻発した。また、すべての実験においてエラーが生じ、メロディを生成できない場合が生じているのも問題点として挙げられる。

しかし、プロンプトの与え方によって、生成されたメロディには大きな差が生じる。本稿で試した音楽理論も、プロンプトの与え方次第では、よりよいメロディを生成することも可能だと考えられる。また、本稿は音高に着目した実験を行ったが、より正確に GPT-4 のメロディ生成能力を検討するためには、音程やリズムなどの音楽理論についても実験をする必要がある。これらについては、今後機会があれば稿を改めて報告したい。

謝辞

本研究は、JSPS 科研費 23K24966、24H00748 の助成を受けた。

参考文献

- [1] 内橋 由佳, 平野 健太郎, 伊丹 誠, 伊藤 紘二, コード進行をベースにして作曲を支援するシステム情報処理学会研究報告音楽情報科学 (MUS), Vol. 1994, No. 16, pp. 1-6, 1994.
- [2] Ang Lv, Xu Tan, Tao Qin, Tie-Yan Liu, Rui Yan, Recreation of Creations: A New Paradigm for Lyric-to-Melody Generation *arXiv:2208.05697*, 2022
- [3] 田中 健, 外山 史, 東海林 健二, 遺伝的アルゴリズムを用いたメロディー進行とリズムの組み合わせによる自動作曲 (2001) 情報処理学会研究報告音楽情報科学 (MUS), Vol. 2001, No. 82, pp. 43-48, 2001.
- [4] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, Christian Frank, MusicLM: Generating Music From Text(2023) *arXiv:2301.11325*, 2023.
- [5] Yongyi Zang, Yixiao Zhang, The Interpretation Gap in Text-to-Music Generation Models(2024) *arXiv:2407.10328*, 2024.
- [6] Shuyu Li, Yunsick Sung, MelodyDiffusion: Chord-Conditioned Melody Generation Using a Transformer-Based Diffusion Model(2023) *Mathematics* 2023, 2023.
- [7] Yi Yu, Abhishek Srivastava, Simon Canales Authors Info, Claims, Conditional LSTM-GAN for Melody Generation from Lyrics *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, Vol. 17, No. 35, pp. 1-20, 2021.
- [8] Chen Zhang, Luchin Chang, Songruoyao Wu, Xu Tan, Tao Qin, Tie-Yan Liu, Kejun Zhang Authors Info, Claims, ReLyMe: Improving Lyric-to-Melody Generation by Incorporating Lyric-Melody Relationships(2022) *MM '22: Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1047-1056, 2022.
- [9] Ruibin Yuan, Hanfeng Lin, Yi Wang, Zeyue Tian, Shangda Wu, Tianhao Shen, Ge Zhang, Yuhang Wu, Cong Liu, Ziya Zhou, Ziyang Ma, Liumeng Xue, Ziyu Wang, Qin Liu, Tianyu Zheng, Yizhi Li, Yinghao Ma, Yiming Liang, Xiaowei Chi, Ruibo Liu, Zili Wang, Pengfei Li, Jingcheng Wu, Chenghua Lin, Qifeng Liu, Tao Jiang, Wenhao Huang, Wenhui Chen, Emmanouil Benetos, Jie Fu, Gus Xia, Roger Dannenberg, Wei Xue, Shiyin Kang, Yike Guo, ChatMusician: Understanding and Generating Music Intrinsically with LLM(2024) *arXiv:2402.16153*, 2025.
- [10] 川口 竜斎, 片寄 晴弘, GPT-4 による対話的音楽生成のためのプロンプトエンジニアリングエンタテインメントコンピューティングシンポジウム 2023 論文集, Vol. 2023, pp. 107-112, 2023.