

研究報告

言語生成 AI を用いたライブコーディングパフォーマンスの検討
Investigation of Live Coding Performance Using Language Generation AI

奥田 知希

Tomoki OKUDA

九州大学大学院芸術工学府

Graduate School of Design, Kyushu University

城 一裕

Kazuhiro JO

九州大学芸術工学研究院

Faculty of Design, Kyushu University

概要

本研究では、テキストベースで音楽の演奏を記述するライブコーディングパフォーマンスと言語生成 AI の親和性に着目し、この2つを組み合わせた新しいライブパフォーマンスの可能性を模索した。具体的なアプローチとして、OpenAI によって開発された言語生成 AI、GPT シリーズにファインチューニングを行い、ライブコーディング言語 TidalCycles のソースコード生成に特化したモデルを作成した。このモデルを用いた演奏における観客の反応を分析し、言語生成 AI の導入がライブコーディングに対する観客の理解に及ぼす影響を考察した。さらに、新たな AI の活用方法として、自身の分身としての AI の活用と、自身を AI による創作プロセスに組み込んだ創作活動を提案する。

1. はじめに

第四次 AI ブームと呼ばれる現代において、人工知能は目まぐるしい進化を遂げ、様々な分野で積極的に AI が活用され始めている。これまでは高価な GPU などのリソースを持つ、限られた研究者だけが使うことができた深層学習技術が、そういったリソースを持たない一般の研究者やクリエイターにとっても実用的なツールとなりつつある。2022 年までは MIDI データの生成が研究の中心となっていた音楽生成 AI であるが、2023 年以降 SunoAI(SunoAI,2023) や MusicGen(Copet,2023) などのテキストから高品質な楽曲を丸ごと生成する AI がブラウザ上で使えるようになった。AI を用いた音楽作品が容易に生み出せるようになった一方で、AI を使った創作活動においては過去の作品の再生産からいかに抜け出すのかという課題が挙げられる。

AI を使い、型破りな音楽表現を生み出した例として、徳井直生による AI DJ Project がある (Tokui,2016)。これは DJ2 人が交代で曲をかけ合う「Back to Back」というスタイルを人間と AI で行うものであり、AI DJ の予測不可能性がパフォーマンスに緊張感をもたらし、人

間の創造性を拡張することに成功した。AI の能力はあくまで訓練データに基づくパターン認識やそのパターンを元にした生成能力に限られる。一方で人間の創造性は新しいコンテキストや状況に対応する能力、無から新たなアイデアを生み出す能力、さらには感情や経験を通じた独自の視点を持つことが可能であり、現状の AI の技術ではまだ再現が難しい。AI を用いて創造的な表現を行うには作品を丸々 AI に出力させるだけでは不十分であり、人間と AI との相互的な関わり合いが必要となる。

本研究では、最も一般利用が進む言語生成 AI とテキストによる記述で演奏を制御するライブコーディングの親和性に着目した。言語生成 AI に丸ごと出力を生成させるのではなく、人間の演奏者が AI の出力を方向付け、時に AI の出力を改変しながら演奏を行い、人間と AI との双方向のコミュニケーションを通して演奏者 1 人ではなし得なかった表現を創出することを試みる。

1.1. 言語生成 AI

2022 年末から大規模言語モデル LLM を活用した AI システムに対する関心は爆発的に高まっており、第四次 AI ブームの様相を示している。OpenAI が 2022 年に公開した AI チャットボット ChatGPT はリリースから 2 ヶ月でユーザー数が 1 億人に達するほど急速に利用が広がり、記事や資料などの文章の生成・校正、プログラミングのソースコード生成・修正など、これまで人間が行っていた様々な仕事の中で活用されている (Wu,2023)。ChatGPT は GPT-3 の後継モデルとして 2022 年 11 月に発表された GPT-3.5 モデルと、その後継モデルとして 2023 年 3 月に発表された最新モデル GPT-4 を元に構築されている (Achiam,2023)。

GPT モデルは Web サイトから収集されたテキストデータを基に訓練されているため、そのデータの収集以降に起こった出来事の情報への対応やウェブから充分

な情報を取得できないマイナーな話題に関する出力を苦手とする。そのような場合においても高い精度の出力を得るための方法としてファインチューニングを用いる。ファインチューニングとは、既存の学習済みモデルを別のデータセットで再トレーニングし、新しいタスクに合わせて微調整することである。GPT-3はファインチューニング無しでも様々な用途の言語生成タスクに用いることができるが、ファインチューニングを行うことで特定のタスクに特化し、より高い精度で言語を生成することができる。本研究ではGPT-3.5モデルにファインチューニングを行い、ライブコーディング言語の生成に特化したGPTモデルを作成する。以下ファインチューニングを行ったGPTモデルをFinetuned GPTモデルと表記する。

1.2. ライブコーディング

ライブコーディングは、プログラム言語を直接操作し、その場で実行することで音や映像を生成するパフォーマンスのことを指す(久保田,2020)。最初にステージ上のスクリーンにエディタを映し出し、ソースコードを編集しながらパフォーマンスをするという形式でライブコーディングを実践したのは、2000年に Alex McLean、Adrian Ward、Dave Griffiths によって結成された Slub というバンドである。2004年には Slub のメンバー Alex McLean と Nick Collins により、TOPLAP というライブコーディングのコミュニティが発足し(TOPLAP,2004)、彼らが発表したマニフェストは TOPLAP コミュニティの考えるライブコーディングの理念や理想を表している。

マニフェストでは、ライブコーディングにおいて、演奏者の創造的なプロセスを観客が直接見て、演奏者の心理にアクセスすることが重要であり、これにより透明性の高い演奏を行うことを推奨している(TOPLAP,2004)。また、ギターの演奏を鑑賞するのにギターの演奏技術を知らなくていいように、ライブコーディングを楽しむのに観客がプログラミングの専門知識を持っている必要はないと示されている。ただし、実際の現場ではライブコーディング言語の技術的な知識を持つ人々には特に魅力的に映るものの、知識がない人々にとっては、その技術的な側面が理解しにくいことがある。このような課題に対処する方法として、視覚的な演出や音楽自体の質の向上を目指す試みが実践されている。

また、本研究ではライブコーディング環境として TidalCycles を取り上げる。TidalCycles は、Alex McLean により開発されたライブコーディング言語であり、Haskell というプログラミング言語のライブラリとして実装される(McLean, 2014)。TidalCycles はパターンの生成に優れていて、複雑なリズムパターンやメロディをリアルタイムに様々な手法で生成したり、変調したりす

る機能が用意されている。TidalCycles は一度実行されたコードが即座に終了するのではなく、生成された音楽的なパターンが繰り返され続ける特性がある。そして、演奏者はこのループを書き換えることで、新たな音楽的なパターンを生成したり、既存のパターンを変化させたりして動的な演奏を繰り返す。

2. 実験

以上の背景を踏まえ、本研究では演奏者の日本語プロンプトによる指示を踏まえ、TidalCycles のコード修正を行う Finetuned GPT モデルを作成する。また、本研究では応答速度が速い GPT-3.5 モデルを採用し、ファインチューニングを行う。データセットとして、ドラムの演奏を記述したソースコードと、それに少し修正を加えたソースコード、そのコード修正の指示を示す日本語のプロンプトという3つの組み合わせを300組用意した。学習を行ったGPT-3.5に対して、「強烈なドラムに」というデータセットに含まれていない日本語プロンプトを用いてコードを修正し、得られた出力を図1、図2に示す。得られたアウトプットは訓練に使用したデータセットには含まれないものであり、モデルが過学習していない良い兆候が見られた。また、出力されたコードは日本語のプロンプトによる演奏者の方向付けが反映されたものであり、データセットに含まれるものと大きく異なるプロンプトではコード修正結果の方向付けがうまくいかない結果となった。これにより、データセットを作成した演奏者自身にしか使いこなせないAIモデルと言えるものとなった。

```
d1
$ stack[
  s "ifdrums:0(3,8)"
]
#shape 0.5
#cps(140/240)
```

図1: 修正前のコード

3. 演奏

前章の実験を踏まえ、演奏者にしか使いこなせないパーソナルなAIを用いて、AIの生成結果を演奏者がデバッグしながらナビゲートすることで演奏を行った。2023年12月21日から23日に行われた配信ライブ Solstice Stream 2023 において本セットアップを用いたライブコーディングパフォーマンスを行った。演奏の流れを図3に、演奏の様子を図4に示す。本演奏では、シンプルなドラムのリズムを表す TidalCycles の

```
d1
$ stack[
  ghost $ s "ifdrums(<3 4>,8)",
  s "cp*8 [hh*2]"
# delay 0.1
# delayt 0.15
# delayfb 0.5
# shape 0.5
# cps (140/240)
```

図 2: 修正後のコード

ソースコードを最初に演奏者が記述した。具体的には bpm150 でガバキックのサンプルを 1 小節に 3 回ずつならし、コンプレッサーをかけて音圧を調整するコードである。このコードに対してどのような修正を加えるか日本語のプロンプトを記述し、Finetuned GPT-3.5 を用いて修正を加えたコードを出力した。コードの修正が終わってから次にどう展開するか考え、日本語のプロンプトを入力し、同様にコード修正を繰り返して行った。Finetuned GPT-3.5 が生成したコードが文法的に誤っている場合、できるだけそのコードを活かして演奏を展開するようにデバッグを行った。演奏者が主体的に演奏に加わったことで、普段なら書かないような記述方法をその場で考えついた場面があった。

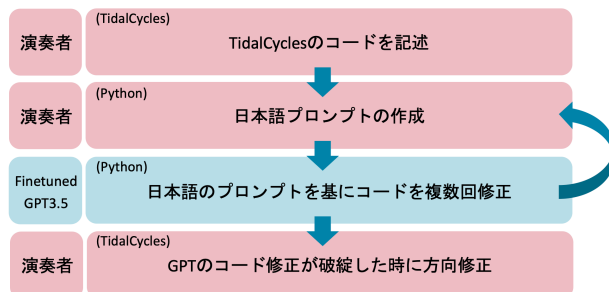


図 3: 演奏の流れ

4. 分析

チャット欄における観客の反応を分析する。最初は GPT をどのように用いるのか疑問を示した聴衆だが、徐々に GPT-3.5 によるコード修正でリズムパターンが複雑になるにつれ、踊る絵文字を使ってその期待感を表現し始めた。コード生成の待ち時間があることや漢字を使った短い文章で GPT に指示を出していることなど、本セットアップの特殊性を面白いコメントが見受けられた。パフォーマンス中盤では他の聴衆に対

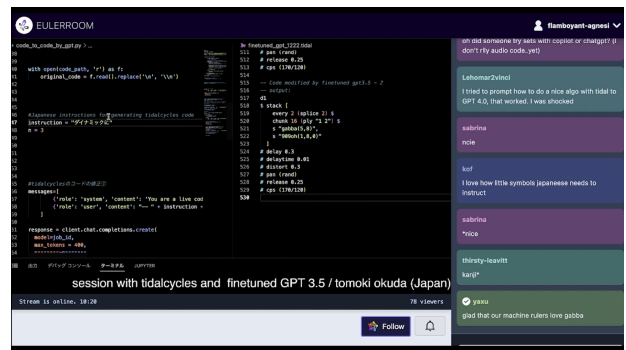


図 4: 演奏の様子

して GPT でライブコーディング言語の生成を試したかどうか問いかけるコメントがあり、それに対して試した上で生成されたコードのクオリティにショックを受けた人のコメントが返ってくるなど、チャット欄で複数人が会話で盛り上がるという配信ライブならではの場面も見られた。最も演奏中にチャット欄が盛り上がったのはガバキックへのコメントで、演奏者が最初にガバキックを鳴らすコードを記述して以降、演奏が終わるまで GPT はガバキックを用いるコードを生成しつづけた。これは日本語プロンプトで他のキックのサンプルを用いるような指示を出さなかったことと、訓練に用いたデータセットにサンプルを突然変更するようなコードが含まれなかったことが原因である。これに対して、観客からは”glad that our machine rulers love gabba”とユーモアを交えた反応があり、このコメントに他の観客から多くの返信が届いた。聴衆の反応を見ると、AI には人格があり、ガバキックを好んで用いているのだと捉え、面白いコメントが多く見られた。実際には GPT の API を呼び出すというコードを実行しているに過ぎないが、聴衆は AI に人格があり、1 人のライブコーダーがコードを記述しているかのような錯覚を感じていたとも考えられる。

5. 考察

言語生成 AI を組み込んだライブコーディングパフォーマンスを通して、「創作者の分身としての AI の活用」と「AI の出力を経験知として捉えること」の 2 つの点に留意することが重要だと考える。その上で「ライブコーディング」における「言語生成 AI」の活用方法を、「創作活動」における「AI」の活用方法と対象を広げても実践可能であることを示す。

5.1. 創作者の分身としての AI の活用

AI の活用方法の 1 つ目は自分自身を表す鏡として AI と向き合うことである。本研究ではもう 1 人の自分

的 AI とのセッションを通じて、自分自身の持つポテンシャルに気付かされる場面があった。生成 AI が出力するソースコードは演奏者自身のソースコードによってファインチューニングされたものであるため、自分自身では記述しないようなコードが生成され、新たなサウンドに出会う場合があったが、出力されたコードを構成するそれぞれのパーツ自体は演奏者が頻繁に記述するコードであった。パーソナルな AI と向き合うことを通して、自分自身をこれまで気づかなかった視点から見つめたり、自分を構成する要素を分解して捉え直したりすることが可能になる。

5.2. AI の出力を経験知として捉えること

AI の活用方法の 2 つ目は AI の生成したものを経験知として捉えることである。つまり、生成 AI の出力に対して他人の創作物と同じように向き合うのではなく、AI の生成の流れの中に自分自身を組み込み、AI と共に自分自身が作り出すという体験を持って向き合うことである。AI の出力をそのまま最終的な出力としてしまうと、演奏者としての楽しさや驚き、発見は徐々に少なくなり、AI がこれまでにないコードの書き方を生成しても、自分のこととして驚きを感じることはできない。本研究では AI の出力したエラーコードを生かして演奏者自身が書き換えるというプロセスを組み込んだことにより、AI が出力したこれまでにないコードに対して、実際に演奏者が手を動かしながら自分の表現の幅を広げる面白さを感じることができた。McLean はインタビューの中で、「パフォーマンス中に新しく学んだことを試し、そこから学ぶことで、新鮮さを感じることができます。」と話している(田所,2018)。Solstice Stream 2023 における演奏では Finetuned GPT-3.5 が演奏者に対し、パフォーマンス中に新しい試みを促す、良き指導者となったように感じた。今後創作活動に当たり前に AI を用いるようになって、最終的なアウトプットまで AI に任せるのではなく、自分自身もその創作の一部に組み込むことで創作の楽しさに立ちかえることができる。

5.3. 「AI」と「創作活動」への拡張

このような考え方を「言語生成 AI」と「ライブコーディング」から「AI」と「創作活動」に広げて考えてみる。例えば、AI を用いて映画を制作することを想定する。製作者は映画の基本的なテーマやプロットのアイデアを AI に入力し、AI がシナリオを生成する。AI との対話の中で自分自身のイメージを分解しながら、制作者が最も描きたい部分やどう描きたいかという部分を明確にしていく。その中で当初なかったような新しい視点やアイデアを得ることが可能である。その後画

像生成 AI を用いて映像を生成し、製作者がそれを編集して最終的な作品とする。AI を用いて生成した映像は事前に考えていた構図や動きと異なる場合もあるだろう。言わば期待外れの生成結果があったとしても、それを活かすような編集を考え、人間の手でエディットすることで、予期しなかった映像表現や展開が生まれる可能性がある。このように、本研究で取り組んだ「言語生成 AI」を「ライブコーディング」に活用する試みの中で得られた考え方は、対象を抽象化して「AI」を「創作活動」に活用する試みにも転用することができると考えられる。本研究ではファインチューニングによってパーソナルな AI を作る実践の中で、自分自身と向き合い、自分の持つポテンシャルに気づく瞬間があったが、自分を構成するパーツを分解して考える姿勢さえあれば ChatGPT のような普遍的な AI においても AI を自分の鏡として向き合う考え方は適用可能である。また、創作活動における最終的なアウトプットに自分自身を置くことは、そこで生まれる作品の質的な問題だけではなく、生成 AI を使っていると忘れがちな、新しいアイデアを自分の手で生み出す喜び、創り出す喜びを再認識することができるという点において重要だ。

6. おわりに

本研究では、言語生成 AI をライブコーディングパフォーマンスに活用する試みを通して、AI を用いた創作活動の可能性について模索してきた。本研究を行った 2022 年、2023 年は生成 AI にとって激動の 2 年間だった。GPT-3 を用いたライブコーディングについて 2022 年 12 月の先端芸術音楽創作学会 JSSA で発表した際には、言語生成 AI の話をしてピンとこない人がほとんどであったが、その後半年の間に爆発的に一般ユーザーにも普及し、バイト先の年配の方に話しても「ああ、ChatGPT みたいなものを使って研究しているのね。」と伝わるほど幅広い人に言語生成 AI が認知されるようになった。AI は技術に精通した一部のアーティストの特権から脱し、幅広いクリエイターによる新たな表現の手段として認識され始めている。

AI 技術の民主化が音楽の創造と享受の方法を根本的に変えるのは時間の問題なのだろうと思う。AI ツールにより音楽制作のハードルが大幅に下がることで、音楽制作に必要な技術的なスキルや経済的なコストが低減する。ライブコーディングはライブコーダーが観客である場合に最も盛り上がるように、技術的な視点や制作の経験のある観客はより豊かな音楽聴取の体験が可能である。また、技術的な視点だけでなく、作曲や演奏の背後にある感情や意図をより深く理解できるようになるだろう。さらに、聴取体験から新しいインスピレーションを得て、その人自身もさらに新しいア

アイデアを作品に反映できる。個人的には AI の民主化により少しでも多くの人があるために作品を作るようになり、音楽文化がより豊かで魅力的なものになることを期待したい。

7. 謝辞

本研究の一部は、日本学術振興会科研費 [JP21H00495] ならびに [JP23H00591] の支援を受け実施された。

8. 参考文献

- SunoAI. (2023). "Bark"GitHub Pages. <https://github.com/suno-ai/bark> (2024/1/15 Accessed)
- Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., and Défossez, A. (2023). Simple and Controllable Music Generation. arXiv preprint arXiv:2306.05284.
- Tokui, N. (2016). "AI DJ Project". <https://naotokui.net/ja/works/ai-djproject-2016-ja/>
- Wu, T., He, S., Liu, J., Sun, S., Liu, K., Han, Q. L., and Tang, Y. (2023). A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5), 1122-1136.
- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., and McGrew, B. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.
- 久保田晃弘. (2020). 『ライブコーディング』アートワード. アートスケープ. 大日本印刷. <https://artscape.jp/artword/index.php/ライブコーディング>. (2024/01/15 Accessed)
- TOPLAP. (2004). "About - TOPLAP". <https://blog.toplap.org/about/>. (2024/1/15 Accessed)
- TOPLAP. (2004). "ManifestoDraft". <https://toplap.org/wiki/ManifestoDraft> (2024/1/15 Accessed)
- McLean, A. (2014). Making programming languages to dance to: Live coding with tidal. In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Functional Art, Music, Modelling & Design* (pp. 63-70).

田所淳. (2018). 演奏するプログラミング, ライブコーディングの思想と実践: Show us your screens. (BNN).

9. 著者プロフィール

奥田 知希 (Tomoki OKUDA)

1998 年生まれ。2022 年より九州大学芸術工学府音響設計コースに進学。福岡を中心にビートメイカー、ライブコーダーとして活動中。

城 一裕 (Kazuhiro JO)

1977 年生まれ。博士 (芸術工学)。英国ニューカッスル大学 Culture Lab、東京藝術大学芸術情報センター [AMC]、情報科学芸術大学院大学 [IAMAS] を経て、2016 年 3 月より九州大学大学院芸術工学研究院音響設計部門准教授。専門はメディア・アート。現在の主なプロジェクトには「Life in the Groove」、「The SINE WAVE ORCHESTRA」、「phono/graph」などがある。



この作品は、クリエイティブ・コモンズの表示 - 非営利 - 改変禁止 4.0 国際 ライセンスで提供されています。ライセンスの写しをご覧になるには、<http://creativecommons.org/licenses/by-nc-nd/4.0/> をご覧頂るか、Creative Commons, PO Box 1866, Mountain View, CA 94042, USA までお手紙をお送りください。