Vol.2021-MUS-130 No.15
Vol.2021-EC-59 No.15
2021/3/16

メロディを対象とした生成Deep Learningモデルの比較

平井 辰典[1,a)]

概要：様々なメディア分野で生成 Deep Learning モデルが提案されており，その生成品質は年々向上している．音楽生成分野における生成モデルの発展についても，「音楽理論を逸脱するような音の生成例が減った」，「文脈を考慮したメロディを生成できている」といった品質の向上は認められると言える．しかし，こと音楽に関しては「より品質の良い生成結果である」ということを評価することが簡単ではない．本稿では，各種生成 Deep Learning モデルの実装を通じて，出力結果の品質を評価するのではなくそれぞれの特徴について客観的な比較を行う．

1. はじめに

VAE や GAN などの生成 Deep Learning モデルの発展により，新たなコンテンツを生成することが可能となっており，その生成品質は日々向上している．コンテンツ生成の分野では，画像の生成や自然言語文章の生成などで大きな成果が上げられているが，音楽の生成も例外ではなく生成品質は向上している．音楽生成の分野では，いわゆる自動作曲技術による出力の質が年々向上しているように見受けられる．

自動作曲研究における大きな課題として，生成結果の評価が困難であることが古くから挙げられている．自動作曲技術に関する研究は，コンピュータが誕生した当初より行われてきているが，その生成結果を正確に比較，評価することは依然として難しい問題である．それため，音楽生成技術がこれまでにどの程度進歩してきたのかといった指標が不明瞭である．そのような中でも，例えばメロディの生成技術に注目すると，近年の最先端技術による生成結果は聴感上「良いメロディ」として受け止められる傾向にあると考えられる．実際に，最新の生成モデルによる生成メロディを実在する作曲者によるメロディと比較した際の実験結果によれば，人間によるメロディ生成に引けを取らない生成結果が得られているという報告もある．そのような結果を根拠に近年のメロディ生成技術は進化していると主張することはできるが，人間が創ったメロディと比べて良かったかどうかを主観評価結果により結論付けることは，必ずしも音楽鑑賞や音楽制作の現場においては本質的な話であるとは言えない．例えば，音楽知識がまったくない人と比べれば，古くから存在するルールベースの自動作曲アルゴリズムでも十分に「良いメロディ」を生成できていたとも言える．つまり，技術の発展を待たずともすでに人間と比較しても遜色のないメロディを作ることはできていたとも考えられ，それを根拠に良いモデルであると主張することは必ずしも十分な主張とは言えない．もちろん近年の生成品質の向上はそのようなレベルには留まっていない．生成モデルの一つである DeepBach の評価結果によると [1]，1272 人の被験者に対するリスニングテストを行った結果，システムによる生成結果がバッハ本人により作曲されたものであると回答された割合が多く，その区別は困難な域にまで達しつつあることが想像できる．一方で，被験者の音楽経験による評価結果の差も大きく，音楽経験が豊富な被験者には生成結果とバッハ本人の曲との区別がついている割合が高かったという．以上のことから，主観評価による聴取実験では，集められた被験者の違いによる評価結果への影響は無視できるようなものではなく，そのような観点からより「良いメロディ」を生成できるモデルとなっているかを公平に評価することは非常に困難な課題であると言える．

さらに問題を難しくしている原因として，近年の生成 Deep Learning モデルの多くは，モデルのアーキテクチャを一箇所だけ変えたり，パラメータを一つだけチューニングするだけで，生成結果が大きく変わってしまうことが挙げられる．極端な場合には，実験者が多くのメロディを生成した中から評価実験に使用する生成結果を恣意的に選んでしまえば，評価結果は簡単に変わってしまう．実験に使用するサンプルはランダムに選んでいたとしても，そもそも，それにいたるハイパーパラメータのチューニングは多くの場合手動で行われるため，そのチューニング作業によ

[1] 駒澤大学
Komazawa University
[a)] thirai@komazawa-u.ac.jp

り評価結果が改善した可能性を否定することはできない．例えば，自然言語処理分野で目を見張る成果を出しているBERT[2] や GPT-3 といったモデルは，一概にどちらのモデルが優れているかを判断することは難しい [3]．それは，それぞれのモデルがタスクによって得手不得手があり，特定のタスクに注目してどちらが優れているかを評価することは可能であるが，学習データの種類や学習時のパラメータといった要因によって結果が簡単に変わってしまうためである．自然言語処理分野の各タスクには，精度を測るための指標や基準が整備されていることが多いため，比較的公平に良し悪しを判断できるが，それでもどちらが良いのかを断言することは困難である．こと音楽に関してはその評価がさらに難しいということは，すべての人の音楽に対する評価が一致することはありえないということから明らかである．

モデルの改善を図っても，より「良いメロディ」を生成できたのかがはっきりしない以上，音楽の生成 Deep Learning モデルに関する研究は，自然言語処理や画像生成分野などですでに成果が出ている技術を，「他分野で良い結果を出した」という根拠の基に音楽生成向けに応用していくという後追いの形になってしまいがちである．

音楽生成技術には二つの進化の方向性がある．一つはよりうまくモデルを学習させるための技術の追求であり，もう一つは単純により「良いメロディ」を生成するための技術の追求である．前者は音楽情報処理に限った話ではなく機械学習技術の進歩に関する話となる．例えば，単純な RNN よりは LSTM などのゲート付きの RNN の方が勾配爆発・消失の問題が起きづらく学習がうまくいくことや，さらに Attention を導入することにより離れた文脈も考慮可能であることなどが当てはまる．これらは学習技術そのものを進歩させる要素であるが，はたして音楽生成において結果をどれだけ左右させるのかは未知数であり，学習技術と「良いメロディ」に寄与するかどうかは別の話である．生成モデルを開発する上では，学習がうまくいった方が良いし，文脈が考慮できないよりはできた方が良いということは疑う余地はないが，一つ一つのモジュールがどれだけ重要なのかを判断することは難しい．

そもそも，生成モデルを実装してメロディの自動生成を行おうと思うと，多くの場合は聴くに堪えない音の連なりが出力されてしまいがちである．より進化した最新の技術では，その聴くに堪えない結果が出力される割合は下がってきていると考えられるが，その割合を改善するためのモデルの調整は開発者の主観により進んでいくことも少なくない．RNN ベースの手法などでは，データセットの楽曲のメロディに続くメロディをどれだけの精度で予測できたかを指標として評価することもあるが，既存楽曲のメロディを正しく予測できるモデルが必ずしも優れたメロディ生成器であるとは限らない．それに加え，聴感上まともな生成結果が出力されたとしても，それが過学習によるもので，学習データの一部をそのまま生成していたからというケースもある上に，仮に過学習ではなくても，タスクに正答できる妥当なメロディを生成した生成器であるということ以上のことは言えない．学習データから獲得したメロディの法則を基に生成すれば「良いメロディ」に聞こえても不思議ではないが，この場合，生成結果の Novelty という観点で良いモデルであるかどうかについては疑問が残る．他にも，ランダムなノイズに基づいてメロディを生成するようなモデルの場合には，新たな結果を生成する度に違った結果を生成することも珍しくなく，生成結果に関して，偶然良いメロディが生成されたという可能性を完全に否定することは難しい．このように，音楽に関する生成モデルの構築は一筋縄にはいかないことが多い．

以上のような現状を踏まえて，本稿では近年成果を挙げている生成 Deep Learning モデルを比較し，それぞれのモデルの特徴を述べながら，どのようなケースでどのモデルの使い勝手が良いのかについて考察する．本稿では，LSTM，Attention，VAE，GAN などの近年の重要技術を取り上げ，実際にそれらのモデルをユーザが活用することを前提としてモデルの比較を行う．本稿には生成結果も掲載するが，結果については重視せず，各モデルを使ってメロディを生成するためにはどのようなデータが必要となるか，どのようなパラメータがあるか，学習自体にどの程度のコストがかかるかなどといった客観的に述べることができる項目を中心に比較を行う．そのような目的から，本研究では各モデルのパラメータのチューニングはほとんど行わない．各モデルのパラメータをチューニングし始めると，何がどの程度寄与して結果が変わったのかを判断することが困難になってしまうからである．当然，十分な学習ができていないケースもあり，本稿に掲載する生成結果は，元の技術を使用したメロディ生成研究の結果（特に，技術のデモを目的として紹介されるようなベストデータ）と比べると劣ってしまうが，それを前提としながらなるべくシンプルな比較をする．

2. 音楽生成に関する近年の研究事例

本章では，音楽生成に関する近年の研究事例について，特に深層学習に関連する技術を取り上げる．音楽生成，自動作曲技術に関する研究は，コンピュータが誕生した当初より行われてきており，これまでに非常に多くの技術提案されてきた．深層学習技術が浸透する以前の音楽生成に関する研究事例については，松原らによるサーベイ論文に詳しくまとめられている [4]．近年，大きな発展を遂げた深層学習ベースの音楽生成技術についても，Briot らにより詳しく調査され，各技術の特徴が事細かにまとめられ，整理・分類されている [5]．その他にも Herremans らによっても，音楽生成システムを目的や機能毎に分類し，どのよ

うにシステムが変遷してきたかについてのサーベイが行われている [6]．これらのサーベイ論文/書籍と本稿との違いとして，本研究では，音楽生成全般ではなく，メロディに特化した深層学習技術に注目した比較を行う．また，特定の論文で発表された特定のネットワークについて述べるというよりも，生成 Deep Learning モデルを語る上で外すことができない重要技術をキーとして，その詳細についての比較をしていく．音楽生成技術について俯瞰的に理解する場合には上記のサーベイを参照されたい．

ここでは，本稿で扱うメロディ生成モデルに関連した研究事例を紹介する．メロディの生成を前提として，特に記号処理ベースの生成モデルに注目する．

Roberts らが提案した MusicVAE は，Recurrent VAE という，VAE の Encoder と Decoder に RNN を用いることで，時系列データを潜在空間表現したモデルである [7]．VAE をベースとしている手法であるため教師データは不要で，学習データをうまく再構成できるかどうかを基に学習可能である．それに加え，モデルの評価自体も再構成品質によって検証可能としている．さらに，リスニングテストによって生成結果の品質を評価，検証している．獲得した潜在空間表現を用いて，ベクトルの補間やベクトルの演算も可能としているため，二つの参照メロディを用意し，一つのメロディを別のメロディに寄せるといった応用が可能である．これは，音楽制作において作成済みのメロディに違ったメロディの要素を付加させたいといったシチュエーションで有用なものになると考えられる．

GAN は，現在の生成 Deep Learning 技術の中でも最も重要な技術であると言っても過言ではないモデルであり，多くの派生技術が提案されている．主に画像生成分野で大きな成功を納めている技術であるが，音楽生成分野でも応用されており，Dong らが提案した MuseGAN はポリフォニックな複数小節の音楽生成を実現している [8]．一方で，GAN ベースの手法では，時系列データの処理には工夫が必要となる．MuseGAN の場合にはピアノロール表現されたデータに対して畳み込み層を使ってモデル化することで，近接するデータの構造を考慮しているが，離れた時刻のコンテキストを考慮することはできない．さらに，あらかじめ決められた長さの小節数でしか学習ができないため，例えば，2 小節毎ずつの音楽生成は可能であるが，前の 2 小節に続くような音楽の生成ができるわけではないため，自由な長さの音楽が生成できない．また，GAN ベースの手法ではランダムなノイズを基に生成を行うため，生成結果の制御は容易ではない．MuseGAN の評価は，著者らが定義したいくつかの指標（空の小節の割合等）を数値化して分析するとともに，ユーザスタディによって実施されている．ユーザスタディは MuseGAN の三種類のモデル間での比較に留まっているため，他の手法との優劣の判断はできない．MuseGAN のようなモデルを実際の音楽制作において活用する場合には，これから音楽を作り始めるという初期状態に MuseGAN で音楽を生成し，それを基に人が編集していって所望の音楽として仕上げていくといった使い方が想定できる．ボタン一つで作曲されるようなイメージとなるため，文字通りの自動作曲と呼べるようなモデルである．生成品質や生成結果の制御性が高まれば，リスナーが気分に応じてその場限りの生成音楽を聴くといった用途で使うことができるかもしれない．

音楽のような時系列データを扱うための代表的なモデルとして RNN が挙げられる．Boulanger-Lewandowski らは，2012 年に RNN と RBM を組み合わせた音楽生成モデルを提案している [9]．RNN ベースのモデルの場合は，モデルによる予測精度を評価することが可能であり，この手法では N-GRAM や GMM+HMM などといった従来の手法との間でその精度を比較している．そのため，予測精度という観点ではモデリングの精度の高さ，つまり技術の進化を示すことができている．一方で前述のように，音楽生成においては必ずしも予測精度が高いモデルが良いモデルであるとは判断できない．

単純な RNN には勾配爆発や勾配消失が起こりやすいという問題があり，それを解決するために現在では LSTM や GRU などといったゲート付きの RNN モデルが主流となっている．しかし LSTM では，注目しているデータから離れた時刻のデータからの影響を十分に考慮することが困難であるという問題がある．それを解決したのが Attention であり，Attention で RNN を置き換えた Transformer というモデルは自然言語処理分野を中心に大きな成果を挙げている．音楽生成において Transformer を用いた OpenAI による MuseNet[10] や Huang らによる Music Transformer[11] は，非常に高品質な音楽生成を実現しており，話題となっている．Music Transformer については，リスニングテストを含む様々な評価を行ったことが報告されているが，MuseNet に関しては，本稿執筆時点では十分な情報が明らかになっていない．このような状態では，MuseNet と Music Transformer のどちらが優れたモデルであるかを判断することはできない．また，Music Transformer のリスニングテストにおいて報告された興味深い結果の一つとして，比較対象に含まれたベースラインの Transformer（Music Transformer とは別のベースラインのモデル）と PerformanceRNN という LSTM ベースのモデルとを比較した際に，Transformer よりも LSTM ベースの手法の方が評価が高かったという結果が出ている．しかも，ベースラインの Transformer の方が，Perplexity という観点では優れていたにも関わらずである．この結果については，Transformer と LSTM を１対１比較した際には統計的に有意な結果とはならなかったものの，メロディ生成モデルの良し悪しを評価することがいかに困難であるかを示すものである．Perplexity が低い Transformer は，データの予測

精度という観点では LSTM よりも優れているが，リスニングテストによる主観的な評価の結果は異なっていた可能性を示唆している．

ここで紹介した研究の多くは，メロディ生成に特化した研究というわけではなく，コードの生成や複数トラックによるポリフォニックな音楽生成までも同時に実現しているものであるが，多くの手法はネットワークの構造や学習データを変えることでメロディ単体の生成にも応用可能である．それぞれの手法には得手不得手があり，異なる制約や学習の条件などもあることから，タスクをメロディの生成に絞ったとしても単純に比較することは困難である．例えば，同じデータセットを対象として，なるべく同じ条件で学習を行い，同じ被験者群を対象として主観評価実験を行うことで生成結果の良し悪しを評価することは可能ではあると考えられる．しかし，生成モデルを活用するシチュエーションによっては生成結果の良し悪しよりも結果の制御性の方が優先されたりすることなども考えられるため，生成結果の優劣をつけることが必ずしも重要とも限らない．

本研究の目的は，ユーザが実際に生成 Deep Learning モデルを活用してメロディを作るというシチュエーションを考え，各モデルの特徴を比較することにある．ネットワークのアーキテクチャが複雑になればなるほど，どの要素が影響したのかを判断することが困難になるため，本稿で比較するモデルは生成 Deep Learning を語る上で重要なモデルに絞ることとする．

3. メロディを対象とした生成 Deep Learning モデルの比較のための前提条件

本章では，既存の大規模な楽曲データセットにより各種生成 Deep Learning モデルを学習し，与えられた入力メロディを基に新たなメロディを生成することで，それらの比較を行う．具体的に比較するメロディ生成のための手法は以下の 6 種類である．

- ランダム生成：4.1 節
- 全結合深層ニューラルネットワーク：4.2 節
- VAE：4.3 節
- Seq2seq（LSTM）：4.4 節
- Attention 付き Seq2seq：4.5 節
- GAN：4.6 節

各モデルの特性上，メロディ生成タスクは同一の条件とはならないが，入力メロディを同一のものとする．ただし，GAN のように，入力メロディを必要としない生成モデルもあるため，すべてのモデルで入力メロディを使うとは限らない．生成するメロディは，モデルの特徴に応じて，入力メロディに続く新たなメロディや，入力メロディに類似するメロディ，入力メロディとは関係ない新たなメロディとする．また，データの表現方法や学習の方法によっては，本稿に記述した以外の方法でもメロディを生成することが可能であるが，あらゆる方法でのメロディ生成について比較することは現実的ではないため，本稿では各モデルに対してそれぞれ一通りのネットワークを構築して学習を行った．

第 1 章で述べたように，パラメータのチューニングによる生成結果への影響をモデル間で公平に測ることは困難である．例えば学習率のチューニングにより大きくパフォーマンスを向上させるモデルもあれば，あまり大きな影響を及ぼさないモデルもあることが想定できる．そのためここでは，各モデルのハイパーパラメータの細かなチューニングや，学習の改善のための新たなモジュールの追加などについては検討しない．なるべく基本的なネットワーク構成で学習及び生成を行うものとする．各モデルの生成結果はモデルの構造を工夫することで改善可能であるが，生成結果の改善がモデルそのものの本質的なものによるものか，パラメータチューニング等がうまくいったことによるものなのかが測りづらい．そのため，ここでは各手法について解説している入門書や各種ツールのチュートリアルに載っているような基本的なネットワーク構成，パラメータをなるべくそのまま用いて学習を行う．これは，過度なパラメータチューニングこそが手法の良し悪しを比較する上でのボトルネックとなりうると考えているからである．将来的には，パラメータチューニングの難易度も含めた比較も検討しているが，本稿では基本的なネットワーク構成での比較のみに留めるものとする．最低限の検討事項として，ロスやパープレキシティといった値が下がるかといった学習の進行を判断する指標について注目し，まったく学習ができていないという状況は避けている．ここで注意が必要な点として，ロスが下がるからといって良いモデルとなっているとは限らないことがある．学習タスクによっては，ロスが下がりづらい場合があったり，そもそもテストデータが十分に用意できない学習の場合には過学習を起こしてしまう可能性もある．

以降に，各モデルでメロディを学習・生成をする上での条件について記述する．

3.1 データセット

Deep Learning において重要となるのは学習データである．なるべく大量の整備されたデータを使用することが望ましい．この学習データの質や量も，最終的な出力を大きく左右する要因となる．

本稿では，The Lakh MIDI dataset[12] から抽出したメロディデータを使用する．The Lakh MIDI dataset は，Web から収集された 176,581 曲分の MIDI ファイルについてのデータセットである．本研究で扱う対象はメロディであるため，このデータセットの中から，メロディのみを抽出する．メロディの抽出は平井と澤田によるメロディの分散表現学習手法 [13] の前処理に則って行い，その結果，10,853

曲分のメロディを取得した．さらに，上述の前処理手法により，取得したメロディの調を推定し，必要に応じて移調をすることで，データセット内のメロディの調の統一を図っている．

3.2 メロディデータの表現方法

メロディは音高と音価との組み合わせからなる音符によって構成されるものであり，音符列によって表現できる．本稿におけるメロディ生成の対象として注目するのは，音符の連なりのみとし，元の MIDI データに付随していたベロシティなどの情報は扱わない．また，メロディは単旋律のもののみをモデル化の対象とする．

メロディをデータとして表現する方法はいくつかあるが，本稿では，音符/休符の種類と音高の組み合わせに対して固有の ID を付与し，それらを語彙として扱う．これにより，自然言語処理分野で扱われる各種手法の適用が容易となる．メロディデータの表現方法としては，ABC 記譜法と同等の表現力を持った表現となっているが，本研究では，MIDI データから直接各音符のノートナンバー（音名）と音価を取得し，配列データとして，休符の情報も含めて保存している．例えば，C4 の四分音符なら「C4:1.0」，D♯5 の八分音符なら「D♯5:0.5」，四分休符なら「R:1.0」のようにテキストとして表現しておき，学習・生成を行う際にはそれらを語彙とした ID（番号）を扱う．

なお，後述する VAE による学習については，入出力のデータの次元数が固定されている必要がある都合から，平井と澤田によるメロディの分散表現学習手法 [13] を適用し，メロディのフレーズを固定長のベクトルに変換して処理を行う．VAE に基づく手法であっても入力の長さを柔軟にさせる方法はあるが，本稿ではシンプルな VAE のみを比較対象とするため，ここではメロディベクトルを採用した．

3.3 比較する上での条件

異なるモデルを公平に比較するためには，なるべく条件を揃える必要がある．しかし，そもそものネットワーク構造や入出力の表現が違うモデルを完全に公平な状態で比較することは困難である．そこで，本稿では生成結果などの特定の要素にはこだわらず，なるべく多面的に比較を行うことにする．そのような中でも，条件を合わせられる箇所については合わせるために，学習を行う際のデータセットについては 3.1 節で述べた内容に，データ表現については VAE 以外の手法については 3.2 節で述べた方法に統一する．

さらに，メロディ生成を行う際のシナリオとして，既存の入力メロディがある状態で，それに基づいて新たなメロディを生成するというシチュエーションを想定する．これは，人が曲作りをしていて行き詰まったときに，機械の力を借りて曲を完成に近づけるような状況を想定したものである．ただし，本稿で比較するすべてのモデルでこの条件に沿ったメロディ生成を行うことができるわけではない．

VAE と GAN については，入力となるノイズを基に完全に新しいメロディを生成することになる．VAE の場合は，このノイズの値として，入力メロディをエンコードした値を用いることも可能で，それをデコードして得られたメロディベクトルは，入力メロディのものと近くなることが期待される．そこで，VAE のメロディ生成のタスクに関しては，入力メロディをエンコード，デコードした結果に最も近い別のメロディを生成することにする．これは，人が作ったメロディの別のパターンを機械に提案してもらうような状況にあたり，例えば，気に入ったメロディに似ている別の新たなメロディを聴いてみたいといったシチュエーションで使用できるようなモデルにあたる．

GAN の場合には，VAE と同様に狙った生成結果が出るような学習が可能な手法も提案されているが，本稿ではシンプルな GAN を対象としたメロディ生成の比較を行うことを目的としているため，ランダムな入力ノイズを基に，まったく新しいメロディを生成する．そのため，GAN の場合には入力メロディとは関係のないメロディ生成が行われる．

GAN 以外の手法で入力メロディとして採用するのは，童謡「きらきら星」のメロディとする．学習したモデルを対象として入力メロディを基にメロディ生成を行い，原則として最初の試行で生成されたメロディを結果の比較に用いることとする．結果の如何によってモデルのチューニングを行うことはしないが，学習を行う際に学習が進んでいるかどうかを基に最低限のパラメータのチューニングは行うものとする．

4. 各モデルの学習/生成方法

以下に，3 章で挙げた 6 つの手法によるモデルの学習の方法及び，メロディを生成する方法について述べる．ここで述べた各種方法によるメロディ生成の比較については 5 章に記述する．

4.1 ランダム生成

本稿は，生成 Deep Learning モデルの比較を行うことを目的としているが，ベースラインとして，生成モデルを使わずに，ランダムにメロディを生成した場合にはどのような結果が得られるのかについても注目する．ランダム生成は，文字通り学習を行わず，ランダムに音符列を生成する方法である．音名の配列（休符を表す R を含む）と十六分音符から全音符までの長さの音価の配列を用意しておき，二つの乱数を生成して配列のインデクスを決定し，それらを組み合わせることでランダムに音符を選ぶ．これを，音符列の長さがあらかじめ決められた長さに達するまで繰り

返すことでランダムなメロディを生成することができる．ランダム生成では，当然実行する度に違ったメロディが生成され，入力としてどのようなメロディを与えようとも影響はない．

どのオクターブ帯のノートが選ばれるかについては，あらかじめ決めておかなければピッチの乱高下が起きてしまう．そのため，生成する音高は C4 のオクターブに限ることとしており，生成されるメロディの帯域は１オクターブに限られている．これについては，学習データ内の音高と音価の頻度分布を基にサンプリングを行うなど，統計的なアプローチを取り入れることでより表現力豊かなランダム生成も可能である．以降，VAE を除くすべてのモデルにおいて，生成される音符の音高は C4 のオクターブ帯に限定するものとする．

4.2 全結合深層ニューラルネットワーク

いわゆる, シンプルな深層ニューラルネットワーク（DNN）でメロディを生成するとどのようにメロディ生成ができるのかについても注目する．ここでは，入力メロディに対してそれに続く新たなメロディを生成するというシチュエーションを想定しているため，ネットワークへの入力はある時刻の音符の種類，出力はそれに続く音符の種類となるようなネットワークを構築した．具体的には，3.2 節に記述した音符の ID 表現を用いて，入力出力ともに one-hot vector を採用した．

入力層と出力層の他に三層の全結合層（各 300 ユニット，活性化関数は sigmoid）を中間層として用意し，3.1 節で紹介したデータセット（データサイズ 3,992,505）を用いて学習を行った．オクターブを限定していることから，音符の遷移のパターンが限られているため，学習開始後にロスの値が下がった後すぐに収束した．

メロディ生成に際しては，入力メロディの最後の音の ID を one-hot vector としてネットワークに入力し，得られた出力の値を次に各音符が出力される確率とみなして，乱数を基に次の一音を選択していった．それを所望の回数繰り返すことでメロディの生成を行った．

4.3 VAE

VAE の学習の場合，入力データをエンコードして潜在空間におけるベクトルとして表現した後に，その潜在ベクトルをデコードすることで入力データを復元する．つまり，入力データと出力データが同じとなるようなネットワークの学習を行うこととなる．そのため，これまでのモデルと同様な音符の ID 表現を入力して，それを復元するような構造で学習しても新しいメロディの生成という観点ではあまり意味のある学習はできない．もちろん MusicVAE[7] のような時系列を考慮できるモデルを構築すればそのような問題も解決可能ではあるが，本稿ではあくまでも基本的なモデルについての検証を行うことを目的としている．

VAE の出力としてメロディを生成するために，入出力はともにメロディを表現するようなベクトルであることが望ましい．そこで，平井と澤田による Melody2vec を適用し，3.1 節のデータセットのメロディをフレーズ毎に分割し，メロディのベクトルを取得した．本稿で検証する VAE の学習は，このメロディのベクトルを対象として行う．具体的には，Melody2vec により取得した 100 次元のメロディベクトルを対象とし，それぞれ二層の全結合層からなるエンコーダ，デコーダを用いて，10 次元の潜在空間が得られるような VAE のネットワークを学習した．学習データに含まれるメロディベクトルのデータサイズは 967,873 であった．学習は 500 エポック行った．

このモデルを使ってメロディを生成する際には，任意の入力メロディに続くメロディを生成するのではなく，任意の入力メロディを入力した際に出力として得られるメロディベクトルに最も近い別のメロディをコサイン類似度を基に算出し，出力することとした．つまり，入力メロディに似ている別のメロディのフレーズを探索し，それらを結合することで新たなメロディを生成する．

4.4 Seq2seq（LSTM）

LSTM によるエンコーダデコーダモデルである Seq2seq は，音符 ID 列を入力し，それに続く次の音符 ID 列を学習するような構造のネットワークを構築した．エンコーダはシンプルに embedding 層と LSTM 層を入力データの数だけ連結し，最後の LSTM セルから出力される隠れ層のベクトル h を出力する．デコーダでは，受け取った h を LSTM セルに入力し，その出力を全結合層へと入力するとともに，次の時刻の LSTM セルに流すような構造となっている．エンコーダから受け取った隠れ層のベクトル h は，各時刻の全結合層と LSTM 層にも入力することによって学習の改善も行っている [14].

入力データ数については，何音の音符を入力として受け取り，何音の音符を出力として出力するかに関するパラメータとなり，同時にどれだけ過去の情報を考慮するかということにもあたる．学習データをトレーニングデータとテストデータに分けて検証した場合，入力データサイズが小さいときほど学習を進めるにつれて精度が向上していくが，それは単に学習タスクが簡単になるだけであり，必ずしも有効な学習ができているとは限らない．入力データサイズを大きく設定すると，ロスが下がっても精度がほとんど上がらないといったこともあった．本稿では，Seq2seq の学習時の入出力のデータサイズは 5 とし，学習は 25 エポック行った．また，embedding の際の次元数は 16，LSTM の隠れ層 h の次元数は 256 として，データサイズは 3,895,847 であった．

メロディの生成に際しては，入力メロディの末尾 5 音符

を入力し，それに続く新たな 5 音を推定し，さらにそれに続く 5 音を順次推定していくといった方法で一連の音符列を生成した．

4.5 Attention 付き Seq2seq

4.4 節の Seq2seq に Attention 機構を追加することで，離れた時刻にも注意を向けられるようなコンテキストベクトルを導入する．上述のネットワークのデコーダの LSTM 層と全結合層との間に Attention 層を挿入し，エンコーダからはすべての時刻の隠れ層 h の値を受け取るような変更を加え，その他は 4.4 節の Seq2seq とまったく同じ条件で学習を行った．

4.6 GAN

GAN の学習は，生成する音符数を固定して音符数分の one-hot vector を入力とした Discriminator と，固定長のノイズベクトル z をシードとして音符数分の one-hot vector を生成する Generator の二つのネットワークが敵対的な学習を行うように実装した．Discriminator と Generator はともに四層の全結合層により構成した．

ノイズベクトル z の次元は 100 とし，ネットワークに一度に入力する音符数は 100 として，データセット内の各楽曲の先頭から 100 音符のみを学習の対象として，音符数が 100 音に満たない楽曲については学習対象から除外した．その結果，10,216 曲分のメロディの学習を行った．今回実装した GAN では，入力メロディを考慮してのメロディ生成はできないため，ランダムな入力ノイズを基にまったく新しいメロディを生成する．そのため，ランダムなメロディ生成と同様に，入力メロディとは関係のないメロディ生成結果が出力される．

5. 各生成モデルの比較

4 章で述べた各種メロディ生成モデルの比較を行う．

5.1 各モデルの特徴の比較

ここでは，各モデルの特徴について，なるべく客観的に検討できる項目を挙げて比較を行う．ランダム生成の特徴については学習を行わないためここでは取り上げず，主に 5.2 節の結果の比較において注目することにする．

5.1.1 ネットワークの構造の比較

ここでは，本稿で取り上げるそれぞれのネットワークの構造に注目した比較を行う．まず，シンプルな DNN は，一般的に生成モデルとして活用されることは少なく，任意の入力を基に，多クラス分類などを行うような識別器として活用することが一般的である．本稿ではあえて最もシンプルなネットワークとして比較対象に入れたが，そのままの形では時系列の考慮ができないなどの欠点があり，本稿では音符に関する bi-gram のマルコフモデルに相当するようなモデル化を行った．そのため，メロディという高度な時系列データを表現するには力不足であると言わざるを得ない．一方で，DNN に関する様々なネットワーク構造やモジュールが提案されており，それらを駆使することで，所望の特徴を持ったネットワークの実現は可能であり，生成 Deep Learning モデルは DNN の派生形であると見なすことができる．例えば，畳み込み層を導入した CNN は主に画像処理分野で成果を挙げているが，音楽のピアノロール表現との相性が良く，局所的な音の変化に関する構造をモデル化できる．ネットワークの設計次第では，作成途中のメロディに合う続きのメロディを生成したり，間が空欄になっているメロディを補間したり，似ている別のメロディを生成したりといった，様々な応用を実現可能である．

VAE は，生成 Deep Learning の発展のきっかけとも言える重要な技術である．入力データをエンコーダによって潜在空間におけるベクトルとして表現し，デコーダによって，そのベクトルから入力データを復元するように学習するネットワーク構造となっている．単純なオートエンコーダと違い，VAE の場合は，潜在空間において距離が近いベクトルから復元されるデータはその特徴も類似したものとなる．そのネットワークの特性上，入力されたメロディがあったときに，それに似ているメロディを探すことはできても，それに続くメロディを生成するといった応用には適していない．VAE を使い，複数のメロディの潜在空間内でのベクトルの補間結果から，メロディのモーフィングを実現することなども可能であり，潜在空間の探索に基づく新たなメロディの生成などの用途に適しているモデルである．

Seq2seq は RNN を使ったエンコーダデコーダモデルで，本稿では LSTM を使って実装している．RNN を使うことによる大きなメリットは，時系列情報が考慮できるということである．学習時には，入出力データの系列長を固定して学習をするが，自然言語処理でいう EOS（end of sentence）のような終了記号を用いれば，柔軟な長さの入出力を表現することも可能である．しかし，Seq2seq では，入力データの長さがどれだけ長くてもエンコーダから出力される隠れ層のベクトルサイズは一定であるため，多くの情報が入力されたときに十分に情報をエンコードすることができないことが考えられる．そこで，Attention 機構を導入することで，すべての LSTM セルから出力される隠れ層のベクトル h をデコーダに渡すことで，すべての入力データを基にエンコードされた情報を基にデコードできる．さらに，その中でも各時刻において注目する要素を臨機応変に変更していくことで，離れた時刻のデータからも十分に影響を受けることができる．本稿で紹介した二つの Seq2seq のモデルの場合は，入出力データの系列長を 5 としたため，音符 5 つ分の入力から，それに続く次の 5 つ分の音符を推定するようなモデルの学習となっている．この

系列長をより長くすることで，さらに離れた時刻の情報も踏まえての推論が可能になるが，その代償として，学習にかかる時間も長くなる．学習の方法によっては，メロディのスタイル変換など，様々な応用が考えられるモデルであり，本稿で紹介したような任意のメロディに続くメロディを生成するような用途では，メロディの打ち込みに行き詰まったクリエータに，それに続く候補を提案するような使い方が想定できる．

GAN は，生成 Deep Learning 技術の発展を急激に加速させた重要なモデルであり，多くの派生モデルが存在する．入力ノイズベクトルを基にメロディを生成する Generator と，生成されたメロディと実在のメロディとを識別するような Discriminator との間で敵対的な学習を行うようなモデルである．メロディ生成の際には，学習済みの Generator にノイズを入力して新たなメロディを生成することになる．この場合，生成結果は 0 から生成されたものであり，音楽制作の現場で実用化することを考えると，それまでの制作作業とは関係なく新しいメロディを生成するという用途では使えるが，ユーザが制作している楽曲に合わせた生成を行うことは難しい．条件を与えて Generator による生成結果を制御する手法はいくつも提案されているが，生成結果の制御が難しいというのが GAN の特徴の一つである．また，メロディ生成というタスクにおける問題点として，データの長さを固定する必要があるという点が挙げられる．画像生成の場合はデータのサイズを固定しても大きな問題にはならないが，メロディの場合に長さが決められてしまうことはあまり好ましいことではない．

以上のようにそれぞれのモデルには特徴があるが，共通して言えることとして，Deep Learning モデルは構造をアレンジすることが可能なため，目的に合わせてある程度のカスタマイズをあする余地があり，それこそが，各手法をベースとして絶えず様々な生成モデルが新たに提案され続けている理由でもある．例えば，ここで紹介した各種モデルは 1 小節の長さを考慮した生成ができないが，入出力のデータ表現方式を変えることで小節毎のメロディ生成をすることもできる．

5.1.2 学習に要する時間の比較

学習に要する時間は，実行環境や，実装の方法に大きく左右されるものである．そのため，正確な比較が困難な要素であるが，本稿では，同じ計算機環境ですべてのモデルを実装していることから，相対的な比較をすることは可能である．なるべく処理時間の差が生じないような実装を意識してはいるが，実装の詳細部分で使っているモジュールの違い等に起因した処理時間の差がイテレーション毎に蓄積されていることも考えられるため，ここに示すデータは厳密なデータではなくあくまでも参考程度の精度であることに注意されたい．また，学習にかかる時間はパラメータによっても大きく異なる．今回示すデータは 4 に記載した条件で 1 エポックの学習を行ったときにかかった時間を基にしている．

表 1 学習にかかった時間の比較（DNN の学習時間に対する比率）

モデル	DNN	VAE	Seq2seq	Attention 付き Seq2seq	GAN
処理時間	1.0	0.0095	2.1	5.8	0.072

深層ニューラルネットワーク（DNN）の 1 エポックの学習に要した時間を 1 としたときの各モデルの学習に要した時間を**表 1** に示す．ネットワークへの入出力のデータの次元数が異なるため，単純な比較はできないことには注意されたい．この結果によると，VAE ⇒ GAN ⇒ Seq2seq ⇒ Attention 付き Seq2seq と，ネットワークの構成要素が増えていくにつれて学習にかかる時間が増えていっていることがわかる．これは想定の範囲内の結果であると言えるが，この結果から言えることとしては，高速に学習を行いたい場合にはなるべくシンプルな構成によりネットワークを構成する必要があるということである．例えば，Attention 機構を追加することによるメリットが学習時間の増加に見合うほど大きくない場合には，Attention 機構をつけないという選択肢も十分に考えられる．特に，学習データを変えながら何度もトライ＆エラーを繰り返して生成モデルの学習を行うようなケースでは，学習時間が短いほど様々な試行ができて有益であるとも考えられる．

5.1.3 学習の難しさについて

ここでは，客観的な指標ではなく，実際に各手法を実装し，学習を行った際に実際に起きた事柄について箇条書き形式で挙げていく．実装の難しさや学習の難しさのような観点が論文に記述されることはあまり一般的ではないが，各モデルを比較する上での一つの観点として述べる．実装の方法などによっては，ここに挙げられた現象が当てはまらないこともあるということには注意されたい．

- 深層ニューラルネットワーク，Seq2seq，Attention 付き Seq2seq の学習時のロスは，最初のエポックで一定程度下がった後，それ以降はあまり下がらずに一定範囲内で振動した
- VAE のロスはエポック数を増やしていっても下がり続けた（500 エポックまで試行）
- GAN の学習はなかなか安定せず，Generator による出力がすべて 0 となるように学習が進んでしまうケースが多かった
- GAN によるメロディの生成結果は，入力ノイズを変えても同じメロディとなってしまうことが多かった

以上のような事柄が本稿で紹介したモデルの実装に際して生じた．GAN における学習の問題については改善方法があると考えられるが，本稿ではその改善自体は目的外とした．

5.2 生成結果の比較

本稿では，メロディの生成結果を比較することは目的としていない．その理由としては，数少ない生成事例を基に，再現性の低い評価法によって比較した場合に，手法間で何らかの有意差があったとしても（もしくはなかったとしても），それが各モデルの優劣を結論付けるには十分な根拠とは言えないと考えているためである．とはいえ，メロディの生成という目的を基にモデルを学習し，実際にメロディを生成した結果を得ているため，ここではあくまでも参考データとしてそれらを示すこととする．ここに示す生成結果は，各種パラメータのチューニングもなしに生成したメロディであるため，各手法の本領を発揮しているとは言い難い結果であるということには注意されたい．また，各モデルによるメロディの生成に際して，乱数を使う要素が入っているモデルでは，実行毎に違った結果が得られる．特に，ランダム生成の場合にそれは顕著であるが，それ以外の手法でも実行毎に得られる結果が異なる．それによって，試行回数を増やしていくことで偶然良いメロディが生成される可能性や，逆に，偶然音楽とは言えないようなメロディが生成されてしまっているという可能性があることにも注意が必要である．

ここでは，試行回数は各モデル 3 回とし，ここに示す結果は最初の 1 回の試行で生成されたメロディである．ランダム生成と GAN による生成以外で用いた入力メロディは童謡「きらきら星」のメロディである．

各モデルによる生成結果を**図 1**〜**図 6** に示す．VAE による生成結果は入力メロディに類似するとされた別のメロディであり，ランダム生成及び GAN による生成結果は入力メロディとは関係のないメロディ，それ以外のメロディは入力メロディに続くメロディとして生成されたメロディである．DNN の場合は入力メロディの最後の 1 音のみを考慮，二種の Seq2seq は最後の 5 音を考慮してそれに続くメロディを生成している．

学習したメロディはすべて調を C メジャーもしくは A マイナーに移調しているものであるため，あまり ♯ がつかない音符が出力されることが自然であるが，モデルによっては多くの ♯ が出現するメロディとなっている．また，リズムに関しても，どのモデルからの出力にも不自然な長音，不規則なリズムなどが多くみられる．これは，単に各モデルのパラメータチューニングが不十分で，出力結果が安定していないことが原因であると考えられるが，メロディの生成モデルを実装しようとした場合に，一度でうまく学習できて理想的な生成結果が得られるような手法はなかなかなく，このように音楽として成立しないようなメロディが生成されることが珍しくないということも示している．

生成するメロディの長さは制御が難しい要素であり，今回は，指定した小節数の長さを超えるまで音符を生成し続けるような処理を行っている．そのため，Seq2seq のように 5 音単位でメロディが出力されるモデルの場合には，生成結果を途中で切っている．GAN の場合も，音符の数は決められるがそれぞれの音符の長さは生成するまでわからないため，ここに示した結果は，生成された 100 音符分のメロディを途中で切ったものである．また，VAE の場合には，メロディの長さとは関係のないメロディベクトルを基に生成を行っているため，出力されるメロディの長さは制御できず，試行毎に長さの異なるメロディが生成される．

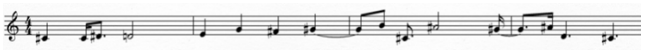

図 1 ランダム生成によるメロディの生成例

図 2 DNN によるメロディの生成例

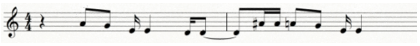

図 3 VAE によるメロディの生成例

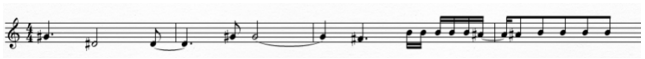

図 4 Seq2seq によるメロディの生成例

図 5 Attention 付き Seq2seq によるメロディの生成例

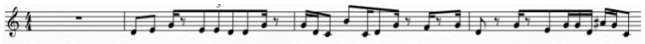

図 6 GAN によるメロディの生成例

5.3 各モデルの音楽制作現場における活用シナリオの検討

各生成モデルは，音楽制作や音楽鑑賞の現場で実際に活用できてこそその真価が発揮されるものである．ここでは，今回実装した各種モデルがどのようなシチュエーションで活用できるかについて検討した．

まず，今回実装した深層ニューラルネットワークの場合，現在の音符情報を基に，次の音符を推論するようなモデルとなっている．その結果が有益であるかは置いておくこととして，実際の楽曲制作の際には，次の音をどうするかに行き詰まった時にモデルによる生成結果を聴いて次の一音を決めるといった活用方法が可能である．

VAE によるメロディ生成の場合，まったく新しいメロディを生成するというよりは，入力メロディに似ている別のメロディを生成するような使用方法を紹介した．これは，リスナーが好みのメロディに似ている別のメロディを探す際に活用できたり，クリエータが自信が考えたものとは違ったパターンのメロディを使いたいときなどに活用可能な方法である．より高度な活用方法としては，VAE の潜在空間を直接探索し，複数のメロディの間に位置するよう

なメロディを探したり，あるメロディから別のメロディに寄せたようなメロディを作りたい時などに活用ができることが考えられる．

Seq2seq の場合は，まとまった音符列を出力として得られるため，メロディ制作に行き詰まったときに，現在作っているメロディに続くメロディの候補を提示してもらうといった使い方ができると考えられる．

GAN によるメロディ生成の場合は，ランダムなベクトルを基にまったく新しいメロディを生成するようなモデルとなるため，メロディを大量に生産したいといったシチュエーションで活用ができると考えられるが，そういった状況はあまり多くはない．今後，GAN によるメロディ生成の精度が向上し，常に聴感上「良いメロディ」が生成できるようになれば，鑑賞用として耐えられるその場限りの音楽を聴き続けられるといったアプリケーションへの活用も考えられる．

6. 音楽生成研究におけるもう一つの方向性

現在の音楽生成研究の主流は生成 Deep Learning モデルに関連したものであるが，一方で，それらの技術を応用していかに創作支援をしていくかを検討するインタフェースについての研究も重要である．Zhou らは，Human-in-the-Loop 最適化により，ユーザが好むメロディを高次元の潜在空間の中から効率的に探索するシステムを提案している [15]．また，Dinculescu らは，MusicVAE のモデルをユーザデータに基づいてパーソナライズする手法 MidiMe を提案している [16]．このような技術により，エンドユーザによる自動作曲結果へのアクセスはより容易になっていくことが予想される．その先には，潜在空間の獲得方法自体をデザインしたいというクリエータが現れることが想定できる．例えば，モデルを構築するための学習データを自身で選びたいといった欲求が生じることは自然な流れであると考えられる．現状では，Magenta.js[17] という JavaScript の API により，Magenta で使用されているモデルを使ったアプリケーションの開発を行うことが可能であるが，使用するためにはプログラミングをする必要があるため，クリエータなどのエンドユーザ向けとは言い難い．DAW ソフトである Ableton Live 向けの VST プラグインとして Magenta.js で提供されている API を使用できるインタフェースとして Magenta Studio[18] が提案されているが，クリエータによる使いやすさを向上させる半面，学習の中身を調整するといった，より細部の調整可能性を捨てるようなデザインとなってしまっている．同様に，クリエータによる生成モデルの活用を可能とするようなツールとして ORB Producer Suite や Flow Machines などが商用化，実用化され始めているが，クリエータ自信が生成モデルそのもののデザインを自由に行えるようなシステムはまだ提案されていない．このように，現状では，自動作曲技術を活用したいクリエータと，自動作曲技術を開発するエンジニアとの間にはギャップが存在する．エンジニアによるモデルのチューニングが結果に与える影響が大きく，生成モデルに対してクリエータがクリエイティビティを反映させる余地はあまりなく，エンジニアが提供するツールの使用に留まっているのが現状であると言える．音楽生成研究の今後の方向性として，生成モデルをいかに活用して人と AI による作曲を実現するかを検討していく必要があり，そのような研究事例も発表されつつある [19]．

ジョン・ケージがプリペアド・ピアノを考案して楽器そのものにまで踏み込んで独自の音楽表現を追求したように，クリエータが独自の生成モデルを作ることを可能とするような仕組みの実現が求められるのではないかと考えている．このような技術の向かう先には，エンジニアが新たな表現を発表するような未来も考えられる．レフ・テルミンによるテルミンの発明や，ロバート・モーグによるシンセサイザの発明のように，新たな表現技術の発明そのものが表現となることも期待でき，生成 Deep Learning 技術はそのようなポテンシャルを秘めている．現状では，新たな生成 Deep Learning のモデルを提案し，そのモデルのパラメータを最適に調整する作業には多大な労力を要する．この作業自体が新たな音楽表現のための活動と言っても過言ではないほどに多くのトライ＆エラーが必要となる．この作業をクリエータ自身が対話的に行えるような生成技術が実現すれば，音楽制作現場でもより広く活用されていくのではないかと考えている．

そのためには，新たな生成モデルを構築する上での様々な課題を解決し，導入の敷居を下げていくことが求められる．現状の音楽生成モデルには，技術的側面，時間的側面，利便性，制御性といった様々な側面における課題がある．例えば一般的なシンセサイザと比べてみるとその違いは大きい．シンセサイザの場合は，パラメータを制御するつまみを捻って鍵盤を押せば，即座に新たな音のフィードバックが得られるが，Deep Learning モデルのパラメータの場合には，つまみを捻って学習をしなおしてようやく新たな音楽が生成される．これらの課題は，より高速な学習が可能なアーキテクチャの登場や効率的かつ直感的なファインチューニング手法の登場を待つこととともに，それらを反映してエンドユーザが使用可能なインタフェースが開発されることによって乗り越えられていくのではないかと期待している．また，複雑な生成 Deep Learning モデルの原理をエンドユーザが理解することは難しいという現状があり，技術的な理解が十分に追い付かない状態でブラックボックス的に使用することとなってしまうことも課題である．AI との共生が話題に登ることも多いが，あくまでもツールを使いこなせなければ AI を活用した創作はできない．他にも，多くの音楽制作ツールでは，各パラメータの値に対して一意に出力が定まる決定的な挙動を示すが，生

成モデルの多くは，確率的な挙動により出力が変化するという問題がある．そのため，現状ではクリエータが意図してパラメータを設定するというよりも，パラメータを変えて偶然出力された気に入った結果を採用するといった使い方になってしまう．生成モデルによる音楽生成の結果がより実用的なものとして音楽制作現場に導入される未来の実現には，結果の不確定性をなるべく排除できるように，今後，ブラックボックスな処理過程の少ないシステムの実現が望まれる．

7. まとめ

本研究では，近年成果を挙げている各種生成 Deep Learning モデルによるメロディ生成についての比較を行った．本稿では深層ニューラルネットワーク，VAE，Seq2seq（LSTM），Attention，GAN を取り上げたが，比較対象となりうる技術は数多く存在し，今回注目したものはその一部に過ぎない．今回の比較は基本的な技術でありながら，近年の生成 Deep Learning モデルにおいてなくてはならない重要な技術のみに焦点を当てたものである．各モデルの良し悪しについての比較まではできていないが，なるべく客観的かつ多面的にそれぞれの特徴や違いをまとめた．今後は，各モデルにより生成されたメロディの良し悪しまでも公平に評価できるような方法について検討していきたい．また，今回評価の対象とはしなかった，最先端の生成技術についても評価ができるような枠組みについても検討していきたい．

「良いメロディ」というものは人によって感じ方が異なり，その定義は困難なものである．そのため，データドリブンのアプローチには限界があり，「音楽理論を逸脱しない」，「無難なメロディ」の生成ができたとしても，それらと「良いメロディ」との間には大きなギャップが存在すると考えられる．そこで重要となるのは，各ユーザが「良いメロディ」だと思えるようなメロディにアクセスするためのインタフェースの実現である．やみくもにメロディ空間を探索しても「良いメロディ」に出会うことは困難だが，近年発展している生成 Deep Learning 技術を活用することで，「良いメロディ」を探索しやすい潜在空間表現の獲得や，所望のメロディに少しでも効率的に近づけるインタフェースの実現が期待できる．今後，そのような技術を実現するために必要な，技術が向かうべき方向性を決める指針についても検討していきたい．

謝辞 本研究は JSPS 科研費 JP 19K20301 の助成を受けたものである．

参考文献

[1] Hadjeres, G., Pachet, F. and Nielsen, F.: DeepBach: a Steerable Model for Bach Chorales Generation, *Proceedings of the 34th International Conference on Machine Learning*, pp. 1362–1371 (2017).

[2] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186 (2019).

[3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D.: Language Models are Few-Shot Learners, *Advances in Neural Information Processing Systems*, Vol. 33, pp. 1877–1901 (2020).

[4] 松原正樹, 深山覚, 奥村健太, 寺村佳子, 大村英史, 橋田光代 and 北原鉄朗: 創作過程の分類に基づく自動音楽生成研究のサーベイ, **コンピュータソフトウェア**, Vol. 30, No. 1, pp. 1_101–1_118 (2013).

[5] Briot, J.-P., Hadjeres, G. and Pachet, F.: *Deep learning techniques for music generation*, Springer (2020).

[6] Herremans, D., Chuan, C.-H. and Chew, E.: A Functional Taxonomy of Music Generation Systems, *ACM Computing Surveys*, Vol. 50, No. 5, pp. 69:1–69:33 (2017).

[7] Roberts, A., Engel, J., Raffel, C., Hawthorne, C. and Eck, D.: A hierarchical latent vector model for learning long-term structure in music, *Proceedings of the 35th International Conference on Machine Learning*, pp. 4364–4373 (2018).

[8] Dong, H.-W., Hsiao, W.-Y., Yang, L.-C. and Yang, Y.-H.: Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, No. 1 (2018).

[9] Boulanger-Lewandowski, N., Bengio, Y. and Vincent, P.: Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription, *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1881–1888 (2012).

[10] Payne, C.: MuseNet. OpenAI, 25 Apr. 2019, openai.com/blog/musenet (2019).

[11] Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Hawthorne, C., Dai, A. M., Hoffman, M. D. and Eck, D.: Music Transformer: Generating Music with Long-Term Structure, *International Conference on Learning Representations* (2018).

[12] Raffel, C.: *Learning-based Methods for Comparing Sequences, with Applications to Audio-to-midi Alignment and Matching*, PhD Thesis, Columbia University (2016).

[13] Hirai, T. and Sawada, S.: Melody2Vec Distributed Representations of Melodic Phrases based on Melody Segmentation, *Journal of Information Processing*, Vol. 27, pp. 278–286 (2019).

[14] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y.: Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, pp. 1724–1734 (online), DOI:

10.3115/v1/D14-1179 (2014).
[15] Zhou, Y., Koyama, Y., Goto, M. and Igarashi, T.: Generative Melody Composition with Human-in-the-Loop Bayesian Optimization, *Proceedings of the 2020 Joint Conference on AI Music Creativity (CSMC-MuMe 2020)* (2020).
[16] Dinculescu, M., Engel, J. and Roberts, A.: MidiMe: Personalizing a MusicVAE model with user data, *orkshop on Machine Learning for Creativity and Design, NeurIPS* (2019).
[17] Roberts, A., Hawthorne, C. and Simon, I.: Magenta.js: A JavaScript API for Augmenting Creativity with Deep Learning, *Joint Workshop on Machine Learning for Music (ICML)* (2018).
[18] Roberts, A., Engel, J., Mann, Y., Gillick, J., Kayacik, C., Nørly, S., Dinculescu, M., Radebaugh, C., Hawthorne, C. and Eck, D.: Magenta Studio: Augmenting Creativity with Deep Learning in Ableton Live, *Proceedings of the International Workshop on Musical Metacreation* (2019).
[19] Huang, C.-Z. A., Koops, H. V., Newton-Rex, E., Dinculescu, M. and Cai, C. J.: AI Song Contest: Human-AI Co-Creation in Songwriting, *International Society for Music Information Retrieval* (2020).