変分オートエンコーダによるドラムからボーカルパーカッションへの楽器音変換と評価

信川凜佳[1,2,a)]　北村優輝士[1,b)]　中村友彦[2,c)]　高道慎之介[3,1,2,d)]　猿渡洋[1,e)]

概要：本稿では，ドラム音からのボーカルパーカッション音の合成に取り組む．ボーカルパーカッションとは，打楽器音を音声で模倣する歌唱法であり，動画共有サービス上で流通する合唱などによく用いられる．ボーカルパーカッション音の合成ができれば，古典的な合唱にとどまらない多様なスタイルに合わせた練習支援に役立つ．ボーカルパーカッションは音声ではあるものの，その音響特性は話声や西洋音楽における歌声と大きく異なる．そのため，従来の音声合成とは異なる扱いが必要となりうる．そこで，我々はボーカルパーカッション音の生成を楽器音合成と解釈し，変分オートエンコーダに基づく realtime audio variational autoencoder（RAVE）を用いた音色変換手法を援用する．提案手法では，RAVE をボーカルパーカッション音で学習し，ドラム音を入力することで対応するボーカルパーカッション音へ変換する．また，リズムや音色の忠実性，ボーカルパーカッションらしさの 3 種の観点で主観評価実験を設計し，提案手法の変換性能を評価した．評価実験では，RAVE の潜在変数を離散化する方法 (VQ-RAVE) と連続値のまま用いる方法 (RAVE) を用い，VQ-RAVE は全評価項目で有意差が示されたものの，ボーカルパーカッションらしさでは RAVE が優れている場合があることが示唆された．

1. はじめに

ボーカルパーカッションは，音声器官（人間が音声を発する際に活用する器官）を通じて打楽器音（ドラム音）を模倣する歌唱法である．この歌唱法は，古くは，西アフリカにおける drum dance のドラム音を伝達する手段 [1] や，北インドの太鼓タブラの演奏を指導する方法 [2] などとして用いられてきた [3]．現代では，主にポピュラー音楽の重唱[*1]アレンジを歌声のみで表現するグループ歌唱形式（いわゆる，アカペラ）の 1 声部として扱われることが多い．アカペラにおいては，リードボーカルやコーラスなどの他声部が言語音（言語を表す音）を発するのに対し，ボーカルパーカッションはドラムの模倣音を発するという違いがある．ドラム音を模するという役割から，ジャズやレゲエなど様々なジャンルにおいて，アカペラ歌唱において主要な役割を果たす．ボーカルパーカッションを含むアカペラ歌唱は，ソーシャルメディアにおいて新たな重唱スタイルとして普及しつつある．

アカペラ歌声合成においては，言語音を発する声部の音声を人工的に合成する方法が提案されている [4]．これは，深層学習を用いて楽譜と歌詞（言語）から言語音を合成する方法である．この方法に加え，ボーカルパーカッション音を人工的に合成することができれば，アカペラに打楽器模擬の声を付与し，様々なジャンルの重唱を表現できると考えられる．しかしながら，ボーカルパーカッションは言語ではなく打楽器を表現するため，既存研究のように言語から合成する方法は妥当ではない．

そこで本研究では，ドラム音からボーカルパーカッション音を合成する方法を提案する．提案法では，変分オートエンコーダ (variational autoencoder; VAE) [5] に基づく realtime audio variational autoencoder (RAVE) [6] を用いた音色変換手法を援用する．RAVE をボーカルパーカッション音で学習し，ドラム音を入力することで対応するボーカルパーカッション音へ変換する．本研究では，合成したボーカルパーカッション音を評価する方法として，リズムと音色の忠実性，ボイスパーカッションらしさを導

[1] 東京大学
Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo, Tokyo 113–8656, Japan
[2] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST), 2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan
[3] 慶應義塾大学
Keio University
[a)] rinka-nobukawa@g.ecc.u-tokyo.ac.jp
[b)] kitamura-makito@g.ecc.u-tokyo.ac.jp
[c)] tomohiko-nakamura@aist.go.jp
[d)] shinnosuke_takamichi@keio.jp
[e)] hiroshi_saruwatari@ipc.i.u-tokyo.ac.jp
[*1] 楽曲が複数の声部により構成されるとき，各声部の歌唱者が 1 人である歌唱．なお，2 人以上の歌唱者が存在する場合は合唱．

入し，提案手法の変換性能を評価する．

2. 関連研究

2.1 ヒューマンビートボックス，ドラム音との関係

ボーカルパーカッションとヒューマンビートボックスは混同されることも多いが，その指す内容は明確に異なる．ボーカルパーカッションは，グループ演奏の中で打楽器のような音声を口で発する技術[*2]であるのに対し，ヒューマンビートボックスは，単独を基本として打楽器に限らない様々な音を発する技術という違いがある [7]．本研究は前者の立場で，打楽器音（ドラム音）と対比させて議論する．

ボーカルパーカッションは，実際のドラムパターンを忠実に再現することが多いとされる [7]．また，ボーカルパーカッションの演奏指導に関する研究では，基本的な 8 ビートのリズムパターンを口で再現する方法が紹介されており，これは実際のドラムパターンを口で模倣する具体的な例となっている [8]．これらの背景から，ボーカルパーカッションは実際のドラムパターンを忠実に模倣することが多く，したがって，ドラム音をボーカルパーカッション音に変換することは，リズム的な一貫性を保ちながら自然な音響的変換を実現する上で合理的であると考えられる．

2.2 ボーカルパーカッションの音響分析

ボーカルパーカッションの音響分析に関する論文がいくつかある．なお，ヒューマンビートボックスのうち打楽器に対応する発音を分析した論文も含めている．

ボーカルパーカッション音はドラム音のような役割を持つため，発声の多くが，乱流雑音源を伴う無声音であると考えられる [9]．そのため，ボーカルパーカッション音の生成モデルの目的関数は，ランダム性の高い位相よりも，振幅に基づくほうが好ましいと言える．後述する目的関数は，振幅スペクトログラムに基づく．

ボーカルパーカッション音は，言語音と同じく音声器官を通じて発声される．しかしながら，その調音や声門の動きは言語音と部分的に異なる．例えば，発声者によってはボーカルパーカッション音を国際音声記号 (international phonetic alphabet; IPA) で記述可能な場合もあるが [10]，熟練した発声者による音は IPA で記述できないとされている [9]．また，ボーカルパーカッションバスドラム音やボーカルパーカッションハイハット音の発声法に関する記述では，言語音を意識しつつも言語音と異なる調音であることが強調されている [11]．以上より，テキスト音声合成のように自然言語や音声記号からではなく，また，言語音からでもない情報（すなわち，前述したドラム音）からボーカルパーカッション音を合成することが好ましいと言える．

2.3 RAVE [6]

本節では，VAE に基づく楽器音生成手法である RAVE を概説する．RAVE の公式リポジトリの実装（公式実装）[*3] のみが公開されている部分もあるため，RAVE そのものが提案された文献 [6] と本稿で用いる RAVE との差異に関しても述べる．

VAE は，ニューラルネットワークを用いた確率的生成モデルの 1 つである．当該モデルでは，D 次元のデータ $\boldsymbol{x} \in \mathbb{R}^D$ の特徴を表す H 次元の潜在変数 $\boldsymbol{z} \in \mathbb{R}^H$ を導入し，$\boldsymbol{z}$ から $\boldsymbol{x}$ への確率的生成過程をニューラルネットワーク（デコーダ）を用いて表現する．デコーダのパラメータの推定問題は，データに関する対数尤度の下限として得られる変分下限の最大化問題に帰着される．この帰着の際に，$\boldsymbol{x}$ から $\boldsymbol{z}$ の変分事後分布のパラメータを推定するニューラルネットワーク（エンコーダ）が導入される．変分下限は，推定された $\boldsymbol{z}$ からデコーダにより生成されるデータの再構成度合いを表現する項（再構成ロス関数）と，変分事後分布と潜在変数の事前分布に対する Kullback–Leibler（KL）ダイバージェンスからなる．典型的には，潜在変数の事前分布と変分事後分布として正規分布が用いられる．また，潜在変数に対してベクトル量子化（vector quantization; VQ）を施す拡張である VQ-VAE [12] も提案されており，文献 [6] では触れられていないものの公式実装にも導入されている．

RAVE の特徴は学習法にある．この学習法は 2 段階からなり，1 段階目ではエンコーダと共にデコーダを VAE の学習法に基づいて学習する．再構成ロス関数として，複数の時間周波数解像度の短時間 Fourier 変換を用いるマルチスケールスペクトルロス関数 [13] を用いる．当該ロスは振幅スペクトログラム領域で計算されることに注意されたい．2 段階目では，敵対的生成ネットワークの学習法 [14] を援用する．具体的には，エンコーダのパラメータを固定した VAE を生成器とみなし，デコーダのパラメータをさらに学習する．敵対的学習のロス関数に加え，マルチスケールスペクトルロス関数と特徴マッチングロス関数 [15] を組み合わせ用いる．文献 [6] では，ヒンジロスを用いた敵対的学習用のロス関数，識別器として単一の畳み込みニューラルネットワークが用いられていた．公式実装では，multi-period discriminator [16] と multi-scale discriminator [17] を併用している．両者ともに複数の時間解像度で識別を行う識別器の構成法である．

3. 提案法

3.1 変換に対する要請

本稿で目指すボーカルパーカッション音の合成手法では，入力として楽譜ではなくドラム音を用いる．ドラム音を用

[*2] それを担う人の意味も含む．

[*3] https://github.com/acids-ircam/RAVE

いることで，楽譜のみでは定まらない打撃時の強弱の情報や，各ドラム楽器（スネアドラム，バスドラム，シンバルなど）の減衰速度の違いなど音響的情報を活用できる．

ドラム音からボーカルパーカッション音への変換を行う際には，ドラム音の何を保存し，変換後の音がどのような性質を持つべきかを明確にする必要がある．そこで，以下の観点から整理を行い 3 つの要請を定義した．

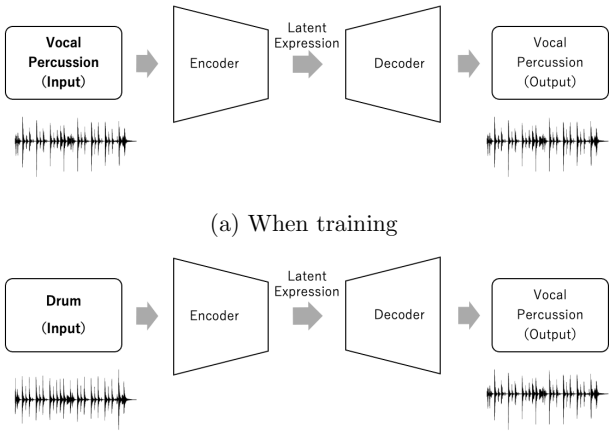
(1) **変換の前後でリズムが保持されること**：専門書 [18] によればヒューマンビートボックスは「音声器官のみを使用して、リズムのあるドラムサウンド、メロディーまたは模倣した楽器を創りだす芸術である。これは、単語の子音または母音だけでなく、非言語音を加えた最先端の歌唱法である」[*4]と定義される．これはヒューマンビートボックスの定義だが，ドラム音に限ればボーカルパーカッションにも適用可能と考えられる．この定義より，ボーカルパーカッションは楽器音と同様にリズムを持つべきだと考えられる．例えば，後述するドラム音–ボーカルパーカッション音の変換においては，ドラム音の入力を受け，そのリズムを保持したボーカルパーカッション音が生成されるべきである．

(2) **変換の前後で音色が一貫していること**：ボーカルパーカッションを楽譜で記号表記する際にはドラム譜を借用した形式をとることが多い．また前述したように，ボーカルパーカッションを音声記号から借用した記号で表す場合もある [11]．これらの事実より，同じ記号で表記される音（例えばボーカルパーカッションのバスドラム音）の音色は一貫しているべきだと言える．故に，ドラム音–ボーカルパーカッション音の変換は一対一に対応することが好ましい．例えば，バスドラム音はボーカルパーカッションのバスドラム音に，スネアドラム音はボーカルパーカッションのスネアドラム音へと対応させる．

(3) **変換後に音声の音響的特徴をもつこと**：ボーカルパーカッションはドラム音の直接的模倣であるが，あくまで音声器官を通じて発生する音声である．故に，ドラム音の音響的特徴を持ちつつも音声のように聞こえなければならない．

3.2 RAVE を用いたボーカルパーカッション音への変換

本稿では，RAVE を用いた音色変換手法をボーカルパーカッション音への変換に適用することを提案する．具体的には，まず RAVE をボーカルパーカッション音で学習し，学習した RAVE モデルにドラム音を入力することで所望のボーカルパーカッション音への変換を行う（図 1）．ここで，学習にはボーカルパーカッション音のみあれば良いことに注意されたい．RAVE は楽器音合成手法であるものの，特定の楽器音で学習したのち他の楽器音を入力することで学習した楽器音へと変換できる [6]．この変換方法では，入力音のオンセットが別の楽器でも保持されやすいことが経験的に知られている．これは，RAVE で用いられるデコーダに波形領域でのエンベロープを直に出力する形式が採用されていることが一因である．そのため，変換後でもリズムが保持されやすいと期待できる．変換前後での音色が一貫しているかに関しては，ドラム音とボーカルパーカッション音の対応関係を直接学習するわけではないため，実験的に確認を行う．変換後にボーカルパーカッション音特有の音響的特徴が含まれるかについては，RAVE でボーカルパーカッション音が適切に表現できるかに関わる．RAVE は打楽器音で比較的よく動作することが知られており，ボーカルパーカッション音に関しても特有の音響的特徴を含む音を表現できることが期待できる．



(a) When training

(b) When converting

図 1: 学習時と変換時の提案法の入出力の違い．

提案法では，ドラム音とボーカルパーカッション音の時間的対応のとれたデータが不要であることも利点である．ボーカルパーカッション音が含まれる代表的なコーパスである jaCappella コーパス [19] には，ドラム音は含まれない．また，楽譜との時間的対応の情報もないため，同一譜面に対する演奏などパラレルデータとなるようなドラム音を作成することは難しい．一方，ボーカルパーカッションの音響信号は含まれるため，ドラム音とボーカルパーカッション音のパラレルデータの不要な提案法はそのまま利用できる．そのため，他の学習データに対する要請も少なく適用が容易である．さらに，RAVE はリアルタイム実行を前提に作られた手法であるため，提案法もその特性を引き継ぎリアルタイム動作を要請する応用にも適用しやすい．

3.3 変換に関する主観評価項目の設計

変換性能を評価する上では，入力音の保持すべき情報が変換後も保持され，音色がボーカルパーカッション音へと適切に変換されることが重要である．本節では次節の主観

[*4] この日本語訳は文献 [7] から引用．

評価実験に合わせ，入力音をドラム音とした場合での設計を考える．3.1 節で示した要請に基づき，以下の 3 つの評価項目を設計した．

(1) **リズムの忠実性：**ドラム音もボーカルパーカッション音も楽曲内のリズムを形作る．本稿で目指す変換は音色に関する変換であるため，変換前後でリズムが一致することが肝要である．そこで，入力したドラム音と変換後のボーカルパーカッション音のリズムが同一であるか否かを，2 値で判定する．

(2) **音色の忠実性：**ドラム音は複数のドラム楽器音から構成される．そのため，単一楽器の音色変換と異なり，ドラム楽器が適切に対応するボーカルパーカッションの発音に割り当てられるかも評価する必要がある．例えば，バスドラムだけでなくスネアドラムの音も，ボーカルパーカッションのバスドラムに対応する音に変換されたとする．この場合，ボーカルパーカッションの音としての自然性は高いものの，変換に伴う楽器の一貫性はなく，提案法で目指す所望の変換ではない．そこで，対応するドラム楽器が適切にボーカルパーカッションの対応する音色へと変換されているか否かを，2 値で判定する．

(3) **ボーカルパーカッションらしさ：**本項目では，ボーカルパーカッション音としての自然性を評価する．ボーカルパーカッション音はドラム音を模倣しているものの，ドラム音にはない吸気音や子音的な音響的特徴が含まれる．このような音声らしさを変換によって付与できているか否かを判定することは重要である．

4. 変換性能に関する主観評価実験

4.1 RAVE の学習

提案法によるボーカルパーカッションへの変換性能の評価のため，主観評価実験を行った．

データセット：データセットとして，日本語のアカペラ重唱曲からなる jaCappella コーパス [19] を用いた．当該コーパスは 10 個のサブセットからなり，各サブセットに 5 曲ずつ 6 声部の重唱アレンジ曲が含まれている．各曲の声部毎の歌唱音源もモノラル音響信号として含まれており，特定声部の単独音源として利用できる．公式のデータ分割方法に従い，各サブセットから 1 曲ずつ合計 10 曲を選択し，ボーカルパーカッションパートの音響信号を検証用データとして用いた．残りの 40 曲分のボーカルパーカッションパートの音響信号は，学習用データとして用いた．全データは 44.1 kHz にリサンプリングし用いた．

学習：前処理として，学習・検証データは曲毎に正規化し，1 s 以上の無音区間の前後で分割した．ここで，無音区間は −60 dBFS 以下となる時間区間を指す．この分割には Python ライブラリ `pydub` を用いた．データ拡張として，ランダムゲイン，ランダムに一部時間区間を無音にするランダムミュート，音響信号処理ソフトウェア `sox` のコンプレッサを用いてダイナミックレンジを変更するランダムコンプレスを用いた．

ニューラルネットワークとしては，変分事後分布として正規分布を採用した VAE によるモデル（RAVE）と，潜在変数を残差 VQ [20] を用いて反復的に離散化する VQ-VAE を用いたモデル（VQ-RAVE）の 2 つを用いた．リアルタイムでの変換にも適用できるように，これらのモデルの主要な構成要素である畳み込み層に対して因果的なフィルタを持つよう制限を加えた．これら 2 つのモデルは RAVE の公式実装に含まれており，そのハイパーパラメータに関してはデフォルト値を用いた．詳細な設定に関しては，RAVE と VQ-RAVE を動作させるための設定ファイル[*5]である付録 A.1 と付録 A.2 を参照されたい．

各モデルは，学習率を 1.0×10^{-3}，勾配の平均と分散に関するモメンタムを $0.5, 0.9$ とした Adam を用いて，$300,000$ エポック学習した．RAVE の第 2 段階の学習に用いる識別器に関しても，同様に Adam で学習した．ただし，学習率は 1.0×10^{-4} とした．

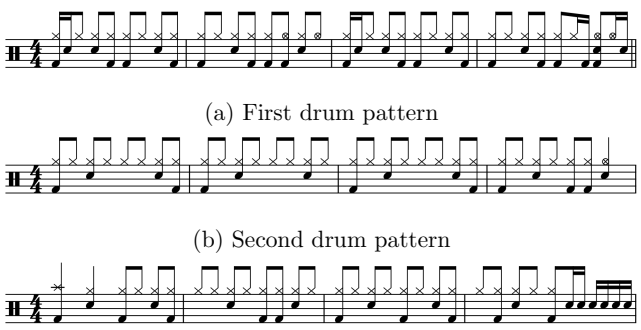
4.2 テストデータの作成

テストデータとなるドラムの音響信号は，ドラム音源ソフトウェア Ezdrummer 3[*6]に収録されているドラムパターンを用いた．用いたドラムパターンの楽譜を図 2 に示す．通常のボーカルパーカッションでは同時に複数楽器を発音できないため，提案法への入力には複数ドラム楽器が同時に発音されていないものをできる限り選定した．なお，ダブルボイスやホーミーなど同時に複数の音を発声する方法もあるが，ボーカルパーカッションとしての奏法の発展的なものとなるため，本稿では除外する．また，テンポやドラムパターンも変換性能への影響しうるため，3 種類のテンポ（80，120，160 BPM）と 3 種類のドラムパターンを用いて評価を行った．テストデータの時間長は，ある程度のドラムパターンの展開が可能な長さとするため，4 分の 4 拍子で 4 小節（6–12 s 程度）となるようにした．こうして作成した合計 9 個のドラム音響信号を学習済みの RAVE と VQ-RAVE にそれぞれ入力し，主観評価に用いるボーカルパーカッションへの変換音を得た．

4.3 被験者への提示方法と評価指標

被験者はクラウドソーシング Lancers[*7]でボーカルパーカッション経験者に回答者を限定して募集した．これは，ボーカルパーカッション経験者に限定することで，ドラム音とボーカルパーカッション音の聴き分け能力に長けた被

[*5] RAVE の公式実装（`https://github.com/acids-ircam/RAVE`）を動作させた際に出力される `config.gin` ファイル．
[*6] `https://www.toontrack.com/product/ezdrummer-3/`
[*7] `https://www.lancers.jp`



(a) First drum pattern

(b) Second drum pattern

(c) Third drum pattern

図 2: テストデータで用いたドラムパターン 3 種の譜面.

表 1: 全評価項目に関する平均値．太字は p 値が 1 %以下であることを表す.

Method	Rhythmic Fidelity	Timbral Consistency	Naturalness as Vocal Percussion
RAVE	**0.74**	0.46	**0.93**
VQ-RAVE	**0.96**	**0.81**	**0.78**

験者を集めるためである．この募集に関して，20〜40 代の男女計 6 名から回答を得た．被験者には，入力ドラム音と変換法 2 種の組み合わせ（計 18 ペア）をそれぞれ提示した．また，実験中に被験者はドラム音，ボーカルパーカッション音を何度も聴き直すことが可能とし，両者の音を最低 1 回は聴取しないと評価ができないようにした．

評価に用いた Web インタフェースを図 3 に示す．評価指標として，3.3 節で提案した指標を用いた．リズムの忠実性に関しては「リズムの忠実性あり」と「リズムの忠実性なし」，音色の忠実性に関しては「音色の忠実性あり」と「音色の忠実性なし」，ボーカルパーカッションらしさに関しては，「ボーカルパーカッションに近い」，「ドラムに近い」のいずれかを選択させた．上記の 3 項目の評価に加えて，各項目の選択理由を記述可能な自由記述欄も用意し，その判断根拠や聴感上のコメントを記述させた．

4.4 実験結果

表 1 に，被験者から得られた各評価項目の平均値を示す．リズムの忠実性と音色の忠実性に関しては「あり」を 1，ボーカルパーカッションらしさに関しては「ボーカルパーカッションに近い」を 1 とした．また，各評価項目に関して片側二項検定を行い p 値を算出した．RAVE モデルでは，リズムの忠実性とボーカルパーカッションらしさの項目で有意差が見られたものの，音色の忠実性では有意差が見られなかった．第一著者が聴取したところ，リズムに関してはよく変換できていたものの，音色面で特に細かいフィルなどを再現できていない傾向があった．このことが音色の忠実性という項目の評価を下げ，またドラムからは程遠いという理由からボーカルパーカッションらしさという項目の評価が上がったと考えられる．これは，jaCappella コーパス内に含まれるボーカルパーカッションに速いパッセージが少ないことに起因する可能性がある．VQ-RAVE モデルでは全項目で有意差が見られ，一定の品質のボーカルパーカッション音が合成できていることが確認できた．これらの結果は，提案法は知覚可能な品質でリズムや楽器の対応関係を保ちつつボーカルパーカッションらしさのある音へと変換できることを示している．

RAVE モデルと VQ-RAVE モデルを比較すると，VQ-RAVE モデルが 3 項目中 2 項目で平均値が高かった．これら 2 つのモデルの差異は主に潜在変数の扱いにあり，RAVE では連続変数として，VQ-RAVE では離散化して扱う．2.2 節で述べた通り，ボーカルパーカッション音はある程度 IPA を用いて記述できるため，離散化が変換性能に有効である可能性がある．一方，ボーカルパーカッションらしさでは RAVE モデルの方が平均評価値は高かった．この結果に関しては，次節で被験者のコメントを通して考察する．

4.5 被験者のコメントに基づく考察

主観評価実験のための Web インタフェースに設けた自由記述欄から，複数名の変換音に関するコメントを得た．RAVE については，以下のようにハイハットやシンバルの響きや，ボーカルパーカッションに含まれる呼吸感に関するコメントが得られた．

- 「金物系 (ハイハット、シンバル) の響く感じが良く表せている」
- 「空気の抜ける音が大きすぎてボーカルパーカッションの音である」
- 「バスドラムの重さが再現出来ておらずいかにも口から発せられている音になっている」
- 「全体的に音色の忠実性が悪く、特にスネアドラムとシンバルの再現性がかなり低いと思います」

また，VQ-RAVE に関しても同様に以下のコメントが得られた．

- 「スネアドラムの響きがよく再現されている」
- 「楽器の違いを表現しつつも、きちんと人の声 (ボーカルパーカッション) に聞こえる」
- 「最後のタム風のスネアドラムの音色は生ドラムとは異なるが、ボーカルパーカッションとしてよく表現されている」
- 「スネア音が内へくぐもってしまっておりボーカルパーカッションの音である」
- 「シンバルの音が、金属的でドラムに近く聞こえる」

このように VQ-RAVE に関してはシンバルに関してドラム音に近く聞こえるというコメントがあり，ボーカルパーカッションらしさの評価値が低くなる要因の 1 つである可能性がある．この要因に関するさらなる調査は今後の課題

図 3: 主観評価実験に用いた Web インタフェース.

とする.

興味深いことに，ボーカルパーカッションらしさについて，「ドラムではなくボーカルパーカッションならではの表現である」という肯定的な評価と，「再現精度が悪いためドラムというよりボーカルパーカッションに近い」という否定的な評価の両方が見られた．前者の意見は，ボーカルパーカッション特有の音響的特徴を付与するという提案法で目指す変換と合致するものであった．一方で，後者は我々が意図していたボーカルパーカッションらしさではないので，より適切な評価を行うため，今後被験者への質問文の設計に工夫を加える必要がある．また，変換元のドラム楽器によっても変換後に重視される音響的特徴が異なることもコメントから示唆されており，どのような特徴が重要であるかの調査を行うことで客観評価指標の作成に役立つと考えられる.

5. おわりに

本稿では，RAVE を用いた音色変換手法を援用し，ドラム音からボーカルパーカッション音へ変換する手法を提案した．また，ボーカルパーカッション音への変換性能を評価するための評価項目に関しても提案した．提案法のバリエーションとして，VAE と VQ-VAE に基づく RAVE モデルを用いて主観評価実験を行い，特に VQ-VAE を用いたモデルに関して知覚可能な品質でリズムや楽器の対応関係を保ちつつボーカルパーカッションらしさのある音へと変換できることを示した．自由記述では，ボーカルパーカッション経験者によるボーカルパーカッションらしさ・ドラムらしさを特徴づける記述が得られ，モデルや評価指標に改善の余地があることが分かった．今後の展望として，楽譜を入力としたボーカルパーカッション音の合成への拡張や，ドラム楽器の同時発音があるドラム音への対応がある.

謝辞： 本研究は，JSPS 科研費 23K18474，21H04900，23K28108，JST 創発的研究支援事業 JPMJFR226V の支援を受けて実施した.

参考文献

[1] J. H. K. Nketia and U. of Ghana. Institute of African Studies., *Our drums and drummers*. Accra: Ghana Pub. House, 1968.

[2] A. D. Patel and J. R. Iversen, “Acoustic and perceptual comparison of speech and drum sounds in the North Indian tabla tradition: An empirical study of sound symbolism,” in *Proceedings of the 15th international congress of phonetic sciences*. Universita Autònoma de Barcelona Barcelona, Spain, 2003, pp. 925–928.

[3] M. Atherton, “Rhythm-speak: Mnemonic, language play or song,” in *Proceedings of the International Conference on Music Communication Science. Sydney, Australia*. Citeseer, 2007, pp. 15–18.

[4] H. Hyodo, S. Takamichi, T. Nakamura, J. Koguchi, and H. Saruwatari, “DNN-based ensemble singing voice synthesis with interactions between singers,” in *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, 2024.

[5] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in *Proceedings of International Conference on Learning Representations*, 2014.

[6] A. Caillon and P. Esling, “RAVE: A variational autoencoder for fast and high-quality neural audio synthesis,” *arXiv preprint arXiv:2111.05011*, 2021.

[7] 河本洋一, “日本におけるヒューマンビートボックスの概念形成 — 世界的な潮流と日本人ビートボクサー “afra” との関わりから—,” **音楽表現学**, vol. 17, pp. 33–52, 2019.
[8] 渡辺興司 and 藤田文子, “音楽科教育におけるヴォーカルパーカッションの演奏指導に関する研究—— 8 ビートの完成を目指して ——,” **茨城大学教育実践研究**, vol. 32, pp. 49–60, 2013.
[9] R. Blaylock, N. Patil, T. Greer, and S. S. Narayanan, “Sounds of the human vocal tract,” in *Interspeech 2017*, 2017, pp. 2287–2291.
[10] M. Proctor, E. Bresch, D. Byrd, K. Nayak, and S. Narayanan, “Paralinguistic mechanisms of production in human “beatboxing”: A real-time magnetic resonance imaging study,” *J. Acoust. Soc. Am.*, vol. 133, no. 2, 2013.
[11] G. Tyte, “Beatboxing techniques,” last viewed January 25, 2025. [Online]. Available: www.humanbeatbox.com
[12] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” in *Proceedings of Advances in Neural Information Processing Systems*, vol. 30, 2017.
[13] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, “DDSP: Differentiable digital signal processing,” in *Proceedings of International Conference on Learning Representations*, 2020.
[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proceedings of Advances in Neural Information Processing Systems*, vol. 27, 2014.
[15] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, “MelGAN: Generative adversarial networks for conditional waveform synthesis,” in *Proceedings of Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, 2019.
[16] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *Proceedings of International Conference on Neural Information Processing Systems*, vol. 33, 2020, pp. 17 022–17 033.
[17] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional GANs,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807.
[18] P. T. Matela, *Human Beatbox - Personal Instrument*. Poland: Merkuriusz Polski, 2014.
[19] T. Nakamura, S. Takamichi, N. Tanji, S. Fukayama, and H. Saruwatari, “jaCappella corpus: A Japanese a cappella vocal ensemble corpus,” in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2023.
[20] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. Han, “Autoregressive image generation using residual quantization,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 523–11 532.

付　　録

A.1 VAE を用いた RAVE の設定ファイル

```
from __gin__ import dynamic_registration
import cached_conv as cc
from cached_conv import convs
import rave
from rave import blocks
from rave import core
from rave import dataset
from rave import discriminator
from rave import model
from rave import pqmf
import torch
import torch.nn as nn

# Macros:
CAPACITY = 96
DILATIONS = [[1, 3, 9], [1, 3, 9], [1, 3, 9], [1, 3]]
KERNEL_SIZE = 3
LATENT_SIZE = 128
N_BAND = 16
NOISE_AUGMENTATION = 0
PHASE_1_DURATION = 1000000
RATIOS = [4, 4, 4, 2]
SAMPLING_RATE = 44100

# Parameters for core.AudioDistanceV1:
core.AudioDistanceV1.log_epsilon = 1e-07
core.AudioDistanceV1.multiscale_stft = @core.
    MultiScaleSTFT

# Parameters for model.BetaWarmupCallback:
model.BetaWarmupCallback.initial_value = 1e-06
model.BetaWarmupCallback.log = True
model.BetaWarmupCallback.target_value = 0.05
model.BetaWarmupCallback.warmup_len = 20000

# Parameters for pqmf.CachedPQMF:
pqmf.CachedPQMF.attenuation = 100
pqmf.CachedPQMF.n_band = %N_BAND

# Parameters for discriminator.CombineDiscriminators
    :
discriminator.CombineDiscriminators.discriminators = \
    [@discriminator.MultiPeriodDiscriminator,
     @discriminator.MultiScaleDiscriminator]

# Parameters for cc.Conv1d:
cc.Conv1d.bias = False

# Parameters for variational/cc.Conv1d:
variational/cc.Conv1d.bias = False

# Parameters for scales/torch.nn.Conv1d:
scales/torch.nn.Conv1d.bias = True
scales/torch.nn.Conv1d.device = None
scales/torch.nn.Conv1d.dilation = 1
scales/torch.nn.Conv1d.dtype = None
scales/torch.nn.Conv1d.groups = 1
scales/torch.nn.Conv1d.padding = 0
scales/torch.nn.Conv1d.padding_mode = 'zeros'
scales/torch.nn.Conv1d.stride = 1

# Parameters for periods/nn.Conv2d:
periods/nn.Conv2d.bias = True
periods/nn.Conv2d.device = None
```

```
periods/nn.Conv2d.dilation = 1
periods/nn.Conv2d.dtype = None
periods/nn.Conv2d.groups = 1
periods/nn.Conv2d.padding = 0
periods/nn.Conv2d.padding_mode = 'zeros'
periods/nn.Conv2d.stride = 1

# Parameters for periods/discriminator.ConvNet:
periods/discriminator.ConvNet.capacity = %CAPACITY
periods/discriminator.ConvNet.conv = @nn.Conv2d
periods/discriminator.ConvNet.kernel_size = (5, 1)
periods/discriminator.ConvNet.n_layers = 4
periods/discriminator.ConvNet.out_size = 1
periods/discriminator.ConvNet.stride = 4

# Parameters for scales/discriminator.ConvNet:
scales/discriminator.ConvNet.capacity = %CAPACITY
scales/discriminator.ConvNet.conv = @torch.nn.Conv1d
scales/discriminator.ConvNet.kernel_size = 15
scales/discriminator.ConvNet.n_layers = 4
scales/discriminator.ConvNet.out_size = 1
scales/discriminator.ConvNet.stride = 4

# Parameters for cc.ConvTranspose1d:
cc.ConvTranspose1d.bias = False

# Parameters for variational/blocks.EncoderV2:
variational/blocks.EncoderV2.adain = None
variational/blocks.EncoderV2.capacity = %CAPACITY
variational/blocks.EncoderV2.data_size = %N_BAND
variational/blocks.EncoderV2.dilations = %DILATIONS
variational/blocks.EncoderV2.keep_dim = False
variational/blocks.EncoderV2.kernel_size = %KERNEL_SIZE
variational/blocks.EncoderV2.latent_size = %LATENT_SIZE
variational/blocks.EncoderV2.n_out = 2
variational/blocks.EncoderV2.ratios = %RATIOS
variational/blocks.EncoderV2.recurrent_layer = None
variational/blocks.EncoderV2.spectrogram = None

# Parameters for blocks.GeneratorV2:
blocks.GeneratorV2.adain = None
blocks.GeneratorV2.amplitude_modulation = True
blocks.GeneratorV2.capacity = %CAPACITY
blocks.GeneratorV2.data_size = %N_BAND
blocks.GeneratorV2.dilations = %DILATIONS
blocks.GeneratorV2.keep_dim = False
blocks.GeneratorV2.kernel_size = %KERNEL_SIZE
blocks.GeneratorV2.latent_size = @core.get_augmented_latent_size()
blocks.GeneratorV2.noise_module = None
blocks.GeneratorV2.ratios = %RATIOS
blocks.GeneratorV2.recurrent_layer = None

# Parameters for core.get_augmented_latent_size:
core.get_augmented_latent_size.latent_size = %LATENT_SIZE
core.get_augmented_latent_size.noise_augmentation = %NOISE_AUGMENTATION

# Parameters for dataset.get_dataset:
# None.

# Parameters for convs.get_padding:
convs.get_padding.dilation = 1
convs.get_padding.mode = 'causal'
convs.get_padding.stride = 1

# Parameters for periods/convs.get_padding:
periods/convs.get_padding.dilation = 1

# Parameters for scales/convs.get_padding:
scales/convs.get_padding.dilation = 1

# Parameters for variational/convs.get_padding:
variational/convs.get_padding.dilation = 1
variational/convs.get_padding.mode = 'causal'
variational/convs.get_padding.stride = 1

# Parameters for discriminator.MultiPeriodDiscriminator:
discriminator.MultiPeriodDiscriminator.convnet = @periods/discriminator.ConvNet
discriminator.MultiPeriodDiscriminator.periods = [2, 3, 5, 7, 11]

# Parameters for discriminator.MultiScaleDiscriminator:
discriminator.MultiScaleDiscriminator.convnet = @scales/discriminator.ConvNet
discriminator.MultiScaleDiscriminator.n_discriminators = 3

# Parameters for core.MultiScaleSTFT:
core.MultiScaleSTFT.magnitude = True
core.MultiScaleSTFT.normalized = False
core.MultiScaleSTFT.num_mels = None
core.MultiScaleSTFT.sample_rate = %SAMPLING_RATE
core.MultiScaleSTFT.scales = [2048, 1024, 512, 256, 128]

# Parameters for blocks.normalization:
blocks.normalization.mode = 'weight_norm'

# Parameters for periods/blocks.normalization:
periods/blocks.normalization.mode = 'weight_norm'

# Parameters for scales/blocks.normalization:
scales/blocks.normalization.mode = 'weight_norm'

# Parameters for variational/blocks.normalization:
variational/blocks.normalization.mode = 'weight_norm'

# Parameters for model.RAVE:
model.RAVE.audio_distance = @core.AudioDistanceV1
model.RAVE.audio_monitor_epochs = 1
model.RAVE.balancer = None
model.RAVE.decoder = @blocks.GeneratorV2
model.RAVE.discriminator = @discriminator.CombineDiscriminators
model.RAVE.enable_pqmf_decode = None
model.RAVE.enable_pqmf_encode = None
model.RAVE.encoder = @blocks.VariationalEncoder
model.RAVE.feature_matching_fun = @feature_matching/core.mean_difference
model.RAVE.gan_loss = @core.hinge_gan
```

```
model.RAVE.input_mode = 'pqmf'
model.RAVE.is_mel_input = None
model.RAVE.latent_size = %LATENT_SIZE
model.RAVE.loss_weights = None
model.RAVE.multiband_audio_distance = @core.
    AudioDistanceV1
model.RAVE.n_bands = 16
model.RAVE.num_skipped_features = 1
model.RAVE.output_mode = 'pqmf'
model.RAVE.phase_1_duration = %
    PHASE_1_DURATION
model.RAVE.pqmf = @pqmf.CachedPQMF
model.RAVE.sampling_rate = %SAMPLING_RATE
model.RAVE.spectrogram = None
model.RAVE.update_discriminator_every = 4
model.RAVE.valid_signal_crop = True
model.RAVE.warmup_quantize = None
model.RAVE.weights = {'feature_matching': 20}

# Parameters for dataset.split_dataset:
dataset.split_dataset.max_residual = 1000

# Parameters for blocks.VariationalEncoder:
blocks.VariationalEncoder.beta = 1.0
blocks.VariationalEncoder.encoder = @variational/
    blocks.EncoderV2
```

A.2 VQ-VAE を用いた RAVE の設定ファイル

```
from __gin__ import dynamic_registration
import cached_conv as cc
from cached_conv import convs
import rave
from rave import blocks
from rave import core
from rave import dataset
from rave import discriminator
from rave import model
from rave import pqmf
from rave import quantization
import torch
import torch.nn as nn

# Macros:
CAPACITY = 96
CODEBOOK_SIZE = 1024
DILATIONS = [[1, 3, 9], [1, 3, 9], [1, 3, 9], [1, 3]]
KERNEL_SIZE = 3
LATENT_SIZE = 128
N_BAND = 16
NOISE_AUGMENTATION = 128
NUM_QUANTIZERS = 16
PHASE_1_DURATION = 200000
RATIOS = [4, 4, 2, 2]
SAMPLING_RATE = 44100

# Parameters for core.AudioDistanceV1:
core.AudioDistanceV1.log_epsilon = 1
core.AudioDistanceV1.multiscale_stft = @core.
    MultiScaleSTFT

# Parameters for model.BetaWarmupCallback:
model.BetaWarmupCallback.initial_value = 0.1
model.BetaWarmupCallback.log = True
model.BetaWarmupCallback.target_value = 0.1
model.BetaWarmupCallback.warmup_len = 1

# Parameters for pqmf.CachedPQMF:
pqmf.CachedPQMF.attenuation = 100
pqmf.CachedPQMF.n_band = %N_BAND

# Parameters for discriminator.CombineDiscriminators
    :
discriminator.CombineDiscriminators.discriminators = \
    [@discriminator.MultiPeriodDiscriminator,
     @discriminator.MultiScaleDiscriminator]

# Parameters for cc.Conv1d:
cc.Conv1d.bias = False

# Parameters for scales/torch.nn.Conv1d:
scales/torch.nn.Conv1d.bias = True
scales/torch.nn.Conv1d.device = None
scales/torch.nn.Conv1d.dilation = 1
scales/torch.nn.Conv1d.dtype = None
scales/torch.nn.Conv1d.groups = 1
scales/torch.nn.Conv1d.padding = 0
scales/torch.nn.Conv1d.padding_mode = 'zeros'
scales/torch.nn.Conv1d.stride = 1

# Parameters for periods/nn.Conv2d:
periods/nn.Conv2d.bias = True
periods/nn.Conv2d.device = None
periods/nn.Conv2d.dilation = 1
periods/nn.Conv2d.dtype = None
periods/nn.Conv2d.groups = 1
periods/nn.Conv2d.padding = 0
periods/nn.Conv2d.padding_mode = 'zeros'
periods/nn.Conv2d.stride = 1

# Parameters for periods/discriminator.ConvNet:
periods/discriminator.ConvNet.capacity = %
    CAPACITY
periods/discriminator.ConvNet.conv = @nn.Conv2d
periods/discriminator.ConvNet.kernel_size = (5, 1)
periods/discriminator.ConvNet.n_layers = 4
periods/discriminator.ConvNet.out_size = 1
periods/discriminator.ConvNet.stride = 4

# Parameters for scales/discriminator.ConvNet:
scales/discriminator.ConvNet.capacity = %CAPACITY
scales/discriminator.ConvNet.conv = @torch.nn.Conv1d
scales/discriminator.ConvNet.kernel_size = 15
scales/discriminator.ConvNet.n_layers = 4
scales/discriminator.ConvNet.out_size = 1
scales/discriminator.ConvNet.stride = 4

# Parameters for cc.ConvTranspose1d:
cc.ConvTranspose1d.bias = False

# Parameters for blocks.DiscreteEncoder:
blocks.DiscreteEncoder.encoder_cls = @blocks.
    EncoderV2
blocks.DiscreteEncoder.noise_augmentation = %
    NOISE_AUGMENTATION
blocks.DiscreteEncoder.num_quantizers = %
```

```
        NUM_QUANTIZERS
93  blocks.DiscreteEncoder.vq_cls = @quantization.
        ResidualVectorQuantization
94
95  # Parameters for blocks.EncoderV2:
96  blocks.EncoderV2.adain = None
97  blocks.EncoderV2.capacity = %CAPACITY
98  blocks.EncoderV2.data_size = %N_BAND
99  blocks.EncoderV2.dilations = %DILATIONS
100 blocks.EncoderV2.keep_dim = False
101 blocks.EncoderV2.kernel_size = %KERNEL_SIZE
102 blocks.EncoderV2.latent_size = %LATENT_SIZE
103 blocks.EncoderV2.n_out = 1
104 blocks.EncoderV2.ratios = %RATIOS
105 blocks.EncoderV2.recurrent_layer = None
106 blocks.EncoderV2.spectrogram = None
107
108 # Parameters for blocks.GeneratorV2:
109 blocks.GeneratorV2.adain = None
110 blocks.GeneratorV2.amplitude_modulation = True
111 blocks.GeneratorV2.capacity = %CAPACITY
112 blocks.GeneratorV2.data_size = %N_BAND
113 blocks.GeneratorV2.dilations = %DILATIONS
114 blocks.GeneratorV2.keep_dim = False
115 blocks.GeneratorV2.kernel_size = %KERNEL_SIZE
116 blocks.GeneratorV2.latent_size = @core.
        get_augmented_latent_size()
117 blocks.GeneratorV2.noise_module = None
118 blocks.GeneratorV2.ratios = %RATIOS
119 blocks.GeneratorV2.recurrent_layer = None
120
121 # Parameters for core.get_augmented_latent_size:
122 core.get_augmented_latent_size.latent_size = %
        LATENT_SIZE
123 core.get_augmented_latent_size.noise_augmentation = %
        NOISE_AUGMENTATION
124
125 # Parameters for dataset.get_dataset:
126 # None.
127
128 # Parameters for convs.get_padding:
129 convs.get_padding.dilation = 1
130 convs.get_padding.mode = 'causal'
131 convs.get_padding.stride = 1
132
133 # Parameters for periods/convs.get_padding:
134 periods/convs.get_padding.dilation = 1
135
136 # Parameters for scales/convs.get_padding:
137 scales/convs.get_padding.dilation = 1
138
139 # Parameters for discriminator.
        MultiPeriodDiscriminator:
140 discriminator.MultiPeriodDiscriminator.convnet =
        @periods/discriminator.ConvNet
141 discriminator.MultiPeriodDiscriminator.periods = [2, 3,
        5, 7, 11]
142
143 # Parameters for discriminator.
        MultiScaleDiscriminator:
144 discriminator.MultiScaleDiscriminator.convnet = @scales
        /discriminator.ConvNet
145 discriminator.MultiScaleDiscriminator.n_discriminators
        = 3
146
147 # Parameters for core.MultiScaleSTFT:
148 core.MultiScaleSTFT.magnitude = True
149 core.MultiScaleSTFT.normalized = False
150 core.MultiScaleSTFT.num_mels = None
151 core.MultiScaleSTFT.sample_rate = %
        SAMPLING_RATE
152 core.MultiScaleSTFT.scales = [2048, 1024, 512, 256, 128]
153
154 # Parameters for blocks.normalization:
155 blocks.normalization.mode = 'weight_norm'
156
157 # Parameters for periods/blocks.normalization:
158 periods/blocks.normalization.mode = 'weight_norm'
159
160 # Parameters for scales/blocks.normalization:
161 scales/blocks.normalization.mode = 'weight_norm'
162
163 # Parameters for model.RAVE:
164 model.RAVE.audio_distance = @core.AudioDistanceV1
165 model.RAVE.audio_monitor_epochs = 1
166 model.RAVE.balancer = None
167 model.RAVE.decoder = @blocks.GeneratorV2
168 model.RAVE.discriminator = @discriminator.
        CombineDiscriminators
169 model.RAVE.enable_pqmf_decode = None
170 model.RAVE.enable_pqmf_encode = None
171 model.RAVE.encoder = @blocks.DiscreteEncoder
172 model.RAVE.feature_matching_fun = @feature_matching
        /core.mean_difference
173 model.RAVE.gan_loss = @core.hinge_gan
174 model.RAVE.input_mode = 'pqmf'
175 model.RAVE.is_mel_input = None
176 model.RAVE.latent_size = %LATENT_SIZE
177 model.RAVE.loss_weights = None
178 model.RAVE.multiband_audio_distance = @core.
        AudioDistanceV1
179 model.RAVE.n_bands = 16
180 model.RAVE.num_skipped_features = 0
181 model.RAVE.output_mode = 'pqmf'
182 model.RAVE.phase_1_duration = %
        PHASE_1_DURATION
183 model.RAVE.pqmf = @pqmf.CachedPQMF
184 model.RAVE.sampling_rate = %SAMPLING_RATE
185 model.RAVE.spectrogram = None
186 model.RAVE.update_discriminator_every = 4
187 model.RAVE.valid_signal_crop = True
188 model.RAVE.warmup_quantize = -1
189 model.RAVE.weights = {'feature_matching': 20}
190
191 # Parameters for quantization.
        ResidualVectorQuantization:
192 quantization.ResidualVectorQuantization.codebook_size
        = %CODEBOOK_SIZE
193 quantization.ResidualVectorQuantization.dim = %
        LATENT_SIZE
194 quantization.ResidualVectorQuantization.num_quantizers
        = %NUM_QUANTIZERS
195
196 # Parameters for dataset.split_dataset:
197 dataset.split_dataset.max_residual = 1000
```