Vol.2022-MUS-133 No.5
2022/1/25

ドメイン敵対的ニューラルネットワークを用いた対数周波数スペクトログラム上の音色変換

深代 勇樹[1]　大村 英史[2]　澤田 隼[2]　桂田 浩一[2]

概要：WaveNet オートエンコーダにドメイン敵対的ニューラルネットワークを導入したモデルを用いて音楽音響信号の曲調や音色などのスタイルの変換を学習する手法が提案されている．しかし，この手法により変換された音楽音響信号は変換前の音楽音響信号に存在した音高を失う傾向がある．本研究では変換する対象をスタイルから音色に限定した音色変換の構築を目指す．音色変換の前後では音楽音響信号の音高は維持されることが望ましい．そこで，変換の前処理として定 Q 変換を導入し対数周波数スペクトログラム間での音色変換をドメイン敵対的ニューラルネットワークに学習させる手法を提案する．対数周波数スペクトログラムは音楽音響信号の音高が特徴としてよく表れる表現である．これにより，先行研究に見られる変換後の音高の消失が起こらない音色変換の実現を目指す．電子的な演奏情報から音楽音響信号を合成することにより得た異なる音色間のパラレルデータを用いた実験と，先行研究の WaveNet オートエンコーダモデルとの間の主観評価の比較実験により，提案する音色変換手法の性能を評価する．

1. はじめに

音楽では，楽曲を異なる楽器構成で演奏するアレンジメントが行われる．楽器構成を替えることにより奏でられる音の音色が変わることで，1 つの楽曲を異なる音色で楽しむことができる．一般の音楽鑑賞者がアレンジメントを聴くにはミュージシャンによってアレンジメントの録音が公開されるのを待つ必要がある．音楽音響信号の音色を自動変換するシステムを構築することで，一般の音楽鑑賞者が主体的に楽曲の音楽音響信号からアレンジメントを作成できるようになり，これまでより能動的に楽曲のアレンジメントを楽しむことができるようになると考えられる．

これまでに，WaveNet オートエンコーダ [1] に音楽音響信号のスタイル変換を学習させる手法が Mor らにより提案されている [2]．この手法では音色だけでなく楽曲の作曲者によって変化する曲調も変換の対象としている．また，この手法では音楽音響信号間での変換をエンドツーエンドで学習している．楽曲の音色を変えるアレンジメントではその前後で音高は維持されることが望ましい．しかし，Mor らの手法には変換前に存在した音高の一部が消失するという問題がある．この問題を防ぐために，音高の特徴をよく表す特徴量間での音色変換を学習するという方法が考えられる．この方法の一つとして対数周波数スペクトログラムを入出力とした音色変換を学習する方法が考えられる．

そこで，本研究では，対数周波数スペクトログラムから計算可能な特徴量のうち対数周波数対数振幅スペクトログラムを用いて，対数周波数対数振幅スペクトログラム上での音色変換を機械学習モデルに学習させる音色変換システムを提案する．これにより，音高の情報を重視した，音色変換の前後での音高の消失が起こらない音色変換システムを構築する．評価実験では，まず，提案手法の音色変換性能を見る客観評価実験を行った．次に，Mor らの手法と提案手法との間の音色変換性能の差を見る主観評価実験を行った．

2. 関連研究

2.1 対数周波数振幅スペクトログラム

音楽音響信号から計算可能な特徴量としてスペクトログラムがある．スペクトログラムは音楽音響信号の周波数成分の系列を時系列で並べたものである．楽音は基本周波数成分とその整数倍の周波数の成分から構成され（調波構造），基本周波数成分は楽音の音高に，すべての周波数成分は楽音の音色に強く影響する．そのため，スペクトログラムは楽音の音高と音色のそれぞれがよく表される音楽音響信号の表現である．スペクトログラムの周波数軸の取り方を対数的にしたものを対数周波数スペクトログラムという．対数周波数スペクトログラムでは調波構造の成分間の対数周波数間隔が音高によらず一定となるため，線形の周

[1] 東京理科大学　理工学研究科
[2] 東京理科大学　理工学部

波数軸を持つスペクトログラムに比べて畳み込みニューラルネットワーク（CNN）の畳み込み層におけるカーネルが音高間で共通する周波数成分の特徴を捉えやすくなることが期待できる．

ここで，音楽音響信号の対数周波数スペクトル $X_{\mathrm{cq}[k]}$ は次式で定義される定 Q 変換 [3] により計算される．

$$X_{\mathrm{cq}}[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} w[k,n]x[n]e^{-\frac{2\pi iQ}{N[k]}n} \qquad (1)$$

ただし，i は虚数単位，$x[n]$ は窓幅 $N[k]$ で切り取られた音楽音響信号の n フレーム目の振幅，$X_{\mathrm{cq}}[k]$ は周波数 f_k に対応する対数周波数スペクトル値である．$k = 0, 1, \cdots, b-1$ は f_k の添字（周波数ビン，b は周波数ビンの数）であり，$f_k = f_0 \cdot 2^{\frac{k}{v}}$ は最小周波数 f_0 より $\frac{k}{v}$ オクターブ高い周波数である．v は 1 オクターブあたりの周波数ビンの分割数である．$N[k] = \frac{Qf_s}{f_k}$ は周波数ビン k に対応する窓幅（f_s は音楽音響信号のサンプリング周波数）であり，$w[k,n]$ は窓幅 $N[k]$ の窓関数である．$Q = \frac{1}{2^{\frac{1}{v}}-1}$ はオクターブの分割数 v により決定する定数である．

音楽音響信号の対数周波数スペクトルを一定のフレーム間隔で計算したものが対数周波数スペクトログラムとなる．対数周波数スペクトログラムの各要素は複素数であり，各要素の絶対値をとったものは対数周波数振幅スペクトログラムと呼ばれる．信号処理で周波数成分を扱う場合は絶対値またはその二乗値をとることが一般的である．また，対数周波数振幅スペクトログラムは 1 チャネルのモノクロ画像のように扱うことができるため，既に研究が進んでいる画像分野の深層学習モデルを適用しやすい．

対数周波数振幅スペクトログラムから音楽音響信号への再合成には Griffin-Lim 法 [4] を適用することができる．Griffin-Lim 法を用いて対数周波数振幅スペクトログラムから音楽音響信号への高品質な再合成を行うには，対数周波数スペクトログラムを計算する際のフレーム間隔を十分小さく設定する必要がある．フレーム間隔が音楽音響信号のサンプリング周波数 f_s に比べて大きいとき，特に高音域での再合成の品質が低下する．しかし，f_k の最大値はサンプリング周波数 f_s により制約されるため，音楽音響信号の周波数成分を十分広い音域に対して得たい場合，サンプリング周波数 f_s を小さくとることは難しい．また，定 Q 変換で用いられる窓関数 $w[k,n]$ の窓幅 $N[k]$ は音楽音響信号のサンプリング周波数 f_s を高く，最小周波数 f_0 を低く設定するほど広くなり，それに伴って定 Q 変換の時間計算量が大きくなる．定 Q 変換は音楽音響信号への再合成の際に繰り返し実行されるため，時間計算量の増加は問題である．

2.2 ドメイン敵対的ニューラルネットワーク

特定のドメインのデータを用いて学習された予測器を異なるドメインのデータに対して適用するドメイン適応という分野がある．予測器の学習に用いられたドメインをソースドメイン，学習された予測器に適用させるドメインをターゲットドメインという．ドメイン敵対的ニューラルネットワーク（DANN）[5] は予測結果の正解ラベルが得られないようなターゲットドメインに対するドメイン適応の手法の一つである．学習モデルは入力データから中間表現へ変換するエンコーダのニューラルネットワーク，中間表現から予測結果を出力するデコーダのニューラルネットワーク，中間表現のドメインを分類する分類器のニューラルネットワークの 3 つで構成される．学習中，エンコーダはソースドメインとターゲットドメインのどちらのデータも入力する．分類器はエンコーダが変換した中間表現がソースドメインのものであるかターゲットドメインのものであるかを分類するよう学習する．それに対し，エンコーダは分類器が中間表現のドメインを分類できないように学習する．これにより，エンコーダはソースドメインとターゲットドメインの区別が付かない中間表現を生成できるようになる．デコーダは中間表現から予測結果を出力するよう学習されるが，エンコーダと分類器の学習により中間表現からドメインを区別する情報が除かれているため，ターゲットドメインのデータから変換された中間表現を受け入れることができる．

2.3 WaveNet オートエンコーダを用いたスタイル変換

Mor らは WaveNet オートエンコーダ [1] にドメイン敵対的ニューラルネットワークの学習手法を導入した音楽音響信号のスタイル変換を提案している [2]．ここで，スタイルには楽曲の音色の他に楽曲の作者のような曲調を変化させる要素も含まれる．Mor らのスタイル変換モデルは 1 つの WaveNet エンコーダと複数の WaveNet デコーダ，1 つの分類器から構成される．スタイル間で共通の WaveNet エンコーダと個々のスタイル専用の WaveNet デコーダの組によりスタイル変換が行われる．WaveNet エンコーダと分類器はドメイン敵対的ニューラルネットワークと同様の学習手法により，エンコーダが出力する中間表現に入力の音楽音響信号のスタイル情報が含まれないように学習される．WaveNet エンコーダと個々の WaveNet デコーダの組は，WaveNet デコーダに対応するスタイルの音楽音響信号の恒等変換を学習する．このとき，WaveNet エンコーダが出力する中間表現はスタイルの情報が除かれているため，WaveNet デコーダは中間表現にスタイルを付与するように学習される．これにより，WaveNet エンコーダと WaveNet デコーダの組がスタイル変換器として学習される．

Mor らの手法により変換された音楽音響信号は変換先の音色特徴がよく現れている．しかし，同時に，変換前の音楽音響信号には存在した音高の一部が変換後の音楽音響信号では消失するという問題がある．

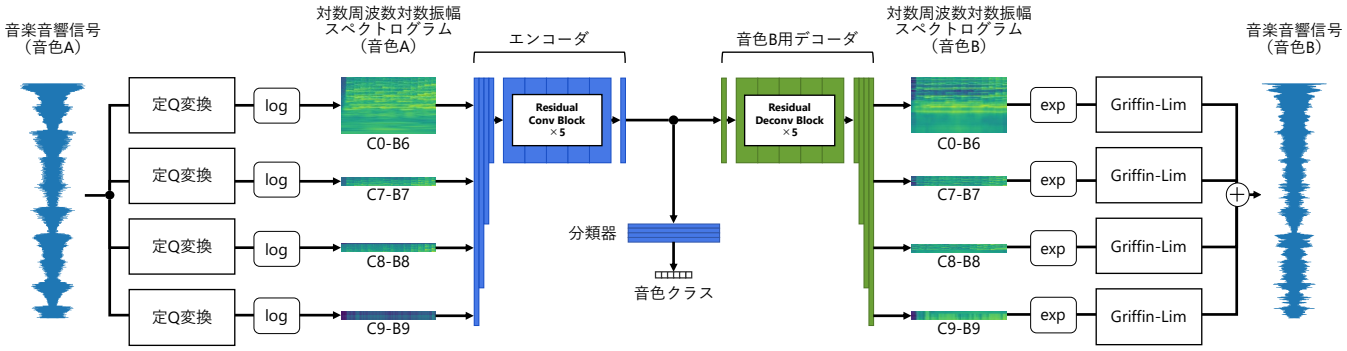

図 1　提案システムの全体

3. 提案手法

3.1 提案システムの概要

本研究では Mor らの手法 [2] をベースに，畳み込みニューラルネットワークベースのドメイン敵対的ニューラルネットワークを構成し，対数周波数対数振幅スペクトログラム上での音色変換を学習する手法を提案する．**図 1** に本研究で提案する音色変換システムの全体を示す．図 1 の一番左がシステムの入力の音楽音響信号である．入力された音楽音響信号は定 Q 変換と対数の計算により 4 つの対数周波数対数振幅スペクトログラムに変換される．中央は提案システムの機械学習モデルであり，異なる音色間での音色変換を 4 つの対数周波数対数振幅スペクトログラム上で行う．変換後の 4 つの対数周波数対数振幅スペクトログラムは指数関数と Griffin-Lim 法の計算の後，加算処理を経て音色変換後の音楽音響信号に再合成される．

先行研究の Mor らの手法と提案手法との間の大きな相違点は機械学習モデルとその入出力である．Mor らの手法では機械学習モデルとして WaveNet オートエンコーダを用いており，その入出力は音楽音響信号である．それに対し，提案手法では機械学習モデルとして残差構造を含む畳み込みニューラルネットワークを用い，その入出力は対数周波数対数振幅スペクトログラムである．

3.2 音楽音響信号から対数周波数対数振幅スペクトログラムへの変換

システムに入力として与えられた音楽音響信号は定 Q 変換 [3] により対数周波数振幅スペクトログラムに変換される．ただし，対数周波数振幅スペクトログラムには第 2.1 節で述べたように，フレーム間隔を大きくとると再合成が難しく，フレーム間隔を小さくとると計算に時間がかかるという欠点がある．そこで，提案するシステムでは入力の音楽音響信号を複数の異なるサンプリング周波数にリサンプリングし，それぞれのリサンプリング後の音楽音響信号から 4 つの音域別の対数周波数振幅スペクトログラムを計算する．4 つの音域は A4 を 440Hz とする科学的ピッチ表記法 [6] を用いて表せば C0-B6，C7-B7，C8-B8，C9-B9 である．各音域に対応させるリサンプリング時のサンプリング周波数，対数周波数振幅スペクトログラムのフレーム間隔，定 Q 変換の最小周波数 f_0 と周波数ビンの数 b を**表 1** に示す．オクターブの分割数 v は音域ごとに共通で $v = 48$ とした．表 1 に示すようにパラメータを決定すると，低音域ではサンプリング周波数が低いため定 Q 変換の窓幅が小さくなり，低音域における定 Q 変換の問題点である時間計算量の増大を防ぐことができる．また，高音域に向けてサンプリング周波数を大きくしながらフレーム間隔は固定値をとることで，高音域の対数周波数振幅スペクトログラムの時間間隔が短くなり（C0-B6 では時間間隔が 10ms であるのに対し，C9-B9 では 1.25ms になる），高音域における再合成時の品質の低下を防ぐことができる．

表 1　リサンプリングと定 Q 変換のパラメータ

音域	C0-B6	C7-B7	C8-B8	C9-B9
サンプリング周波数 [Hz]	6400	12800	25600	51200
フレーム間隔	64	64	64	64
最小周波数 f_0 [Hz]	16.351	2093.0	4186.0	8372.0
周波数ビン数 b	336	48	48	48

ニューラルネットワークではその入出力の値域を正規化することで学習が安定化することが多い．ここで，対数周波数振幅スペクトルの要素の最大値を考える．音楽音響信号はサンプリングの際に $[-1.0, 1.0]$ の区間にクリッピングされることをふまえて，複素音楽音響信号 $x[n]$ について $|x[n]| \leq 1$ とする．(1) 式の定 Q 変換の定義式の総和の内側の項の絶対値について，$w[k,n] \geq 0$，$|x[n]| \leq 1$ のとき，次の不等式が成り立つ．

$$\left| w[k,n]x[n]e^{-\frac{2\pi iQ}{N[k]}n} \right| \leq w[k,n]$$

このことから，すべての n に対して $\left| x_{\max}[n]e^{-\frac{2\pi iQ}{N[k]}n} \right| = 1$ となる $x_{\max}[n]$ に対して定 Q 変換を適用すれば，その対数周波数振幅スペクトルの各要素の値が任意の音楽音響信号の対数周波数振幅スペクトルの最大値となることがわかる．そして，$x_{\max}[n]$ は次式で与えられる．

$$x_{\max}[n] = e^{\frac{2\pi iQ}{N[k]}n} \cdot e^{i\theta}$$

ただし，θ は位相を表す定数であり，$x_{\max}[n]$ の位相が対数周波数振幅スペクトルの最大値の計算に影響しないことを表す．これにより計算される $x_{\max}[n]$ の対数周波数振幅スペクトル $X_{\max}[k]$ を用いて次式により計算される正規化された対数周波数対数振幅スペクトログラム $X_{\mathrm{logcq,norm}}[k,n]$ の値域は $[-0.5, 0.5]$ となる．

$$X_{\mathrm{logcq,norm}}[k,n] = -\frac{1}{\log\varepsilon}\log\max\left(\frac{X_{\mathrm{cq}}[k,n]}{X_{\max}[k]},\varepsilon\right) + 0.5$$

ただし，ε は対数の計算による値の発散を防ぐための微小な値，$\max(\cdot,\cdot)$ は 2 つの引数のうちより大きい値を返す関数である．

対数周波数振幅スペクトログラムの対数をとる理由は，次節で説明する音色変換モデルが対数周波数振幅スペクトログラムの上では学習が進行せず，より小さい振幅値に対して値の変動が鋭敏になる対数周波数対数振幅スペクトログラムを用いたところ学習の進行が確認できたためである．

3.3 対数周波数対数振幅スペクトログラム上の音色変換

提案手法では正規化した対数周波数対数振幅スペクトログラム上での音色変換モデルを構築する．対数周波数対数振幅スペクトログラムが 1 チャネルの画像として扱うことができることから，画像分野で多く利用されている畳み込みニューラルネットワーク（CNN）をベースとした音色変換モデルとして用いる．提案手法の音色変換モデルは CNN ベースの単一のエンコーダ，CNN ベースの複数のデコーダ，CNN ベースの単一の分類器から構成される．

提案手法の音色変換モデルで用いるエンコーダ，デコーダ，分類器の構成をそれぞれ**表 2**，**表 3**，**表 4** に示す．各表の第 1 列にはモデルの入出力を区別するために用いる名称と，モデルの複数層をまとめて区別するために用いる名称を示している．表の第 2 列以降は各ニューラルネットワークの層を示している．層の一部では末尾に “, R” または “, T” を付している．これは層の出力の直前に適用する活性化関数を示しており，‘R’ は ReLU，‘T’ は tanh 関数を表す．表の途中で列が結合する箇所は上部の層からの入力が層の処理によって一つの出力にまとめられたことを示す．対称的に，表の途中で列が分岐する箇所は上部の層の出力が複数の並列する層の入力として用いられることを示す．一部の行では特徴量のサイズを一つ組または三つ組で示しており，左から順にチャネル数，対数周波数軸方向の長さ，時間軸方向の長さを表す．ただし，T は C0-B6 の音域における入力の時間軸方向の長さであり，任意の時間長 T の対数周波数対数振幅スペクトログラムをエンコーダに入力することができる．

表 5 に音色変換モデルに現れる畳み込み層および転置畳み込み層のパラメータを示す．C_{out} の列は出力チャネル数を示している．畳み込み層および転置畳み込み層の次元はすべて 2 次元であり，カーネル，ストライド，パディングのサイズを（対数周波数軸方向）×（時間軸方向）の順で示している．層名に ‘*’ を付していないものが畳み込み層，層名に ‘*’ を付したものが転置畳み込み層である．

エンコーダの結合部における Concat 層では 2 つの特徴量を対数周波数軸方向に沿って結合する．結合は 3 回行われ，4 つの音域の特徴量は 1 つの特徴量にまとめられる．残差部の残差構造は複数の畳み込み層と加算処理からなる構造であり，ResNet[7] で初めて用いられた．残差構造の入力は出力まで直接スキップされ，残差構造中に並列する畳み込み層の出力と加算される．表中ではこの加算処理を Add で示している．残差部の残差構造は並列する 1 層の線形補間（LinearInterpolation）と 4 層の畳み込み層（Conv3x5, Conv4x5）から構成される．

表 2 エンコーダの構成

音域	C0-B6	C7-B7	C8-B8	C9-B9
入力	$(1, 336, T)$	$(1, 48, 2T)$	$(1, 48, 4T)$	$(1, 48, 8T)$
結合部	First1x1	First1x1, R	First1x1, R	First1x1, R
				Down3x4, R
			Concat	
			Down3x4, R	
		Concat		
		Down3x4		
	Concat			
結合後	$(64, 480, T)$			
残差部	LinearInterpolation			Conv3x5(s1x1, p1x2), R
				Conv4x5(s2x1, p1x2), R
				Conv3x5(s1x1, p1x2), R
				Conv4x5(s2x1, p1x2)
	Add			
	LinearInterpolation			Conv3x5(s1x1, p1x2), R
				Conv4x5(s2x1, p1x2), R
				Conv3x5(s1x1, p1x2), R
				Conv4x5(s2x1, p1x2)
	Add			
	LinearInterpolation			Conv3x5(s1x1, p1x2), R
				Conv4x5(s2x1, p2x2), R
				Conv3x5(s1x1, p1x2), R
				Conv4x5(s2x1, p1x2)
	Add			
	LinearInterpolation			Conv3x5(s1x1, p1x2), R
				Conv4x5(s2x1, p1x2), R
				Conv3x5(s1x1, p1x2), R
				Conv4x5(s2x1, p1x2)
	Add			
	LinearInterpolation			Conv3x5(s1x1, p1x2), R
				Conv4x5(s2x1, p1x2), R
				Conv1x5(s1x1, p0x2), R
				Conv1x5(s1x1, p0x2)
	Add			
最終層	Conv1x1(s1x1, p0x0), T			
中間表現	$(64, 1, T)$			

表 3 デコーダの構成

中間表現	$(64, 1, T)$			
初期層	Conv1x1(s1x1, p0x0)			
残差部	LinearInterpolation	Deconv1x5(s1x1, p0x2), R		
		Deconv1x5(s1x1, p0x2), R		
		Deconv4x5(s2x1, p1x2), R		
		Deconv3x5(s1x1, p1x2)		
	Add			
	LinearInterpolation	Deconv4x5(s2x1, p1x2), R		
		Deconv3x5(s1x1, p1x2), R		
		Deconv4x5(s2x1, p1x2), R		
		Deconv3x5(s1x1, p1x2)		
	Add			
	LinearInterpolation	Deconv4x5(s2x1, p1x2), R		
		Deconv3x5(s1x1, p1x2), R		
		Deconv4x5(s2x1, p2x2), R		
		Deconv3x5(s1x1, p1x2)		
	Add			
	LinearInterpolation	Deconv4x5(s2x1, p1x2), R		
		Deconv3x5(s1x1, p1x2), R		
		Deconv4x5(s2x1, p1x2), R		
		Deconv3x5(s1x1, p1x2)		
	Add			
	LinearInterpolation	Deconv4x5(s2x1, p1x2), R		
		Deconv3x5(s1x1, p1x2), R		
		Deconv4x5(s2x1, p1x2), R		
		Deconv3x5(s1x1, p1x2)		
	Add			
分割前	$(64, 480, T)$			
分割部	SliceL336	SliceH144		
	Last1x1, T	Up3x4, R		
		SliceL48	SliceH96	
		Last1x1, T	Up3x4, R	
			SliceL48	SliceH48
			Last1x1, T	Up3x4, R
				Last1x1, T
音域出力	C0-B6 $(1, 336, T)$	C7-B7 $(1, 48, 2T)$	C8-B8 $(1, 48, 4T)$	C9-B9 $(1, 48, 8T)$

表 4 分類器の構成

中間表現	$(64, 1, T)$
分類器	Conv1x1(s1x1, p0x0), R
	Conv1x1(s1x1, p0x0), R
	Conv1x1(s1x1, p0x0)
	ClassifierLast1x1
	Global Average Pooling
クラス分類	(K)

デコーダはエンコーダの構成を反転し，カーネルサイズが 1×1 であるもの以外の畳み込み層をすべて転置畳み込み層 [8] に置き換えたものである．ただし，残差構造の加算処理は各残差構造の最終層のままである．分割部の “Slice” で始まる層はエンコーダの Concat 層の逆の処理であり，1 つの特徴量を対数周波数軸方向に分割する．“Slice” に続く大文字の L と H はそれぞれ低音域と高音域を表し，さらに続く数字は分割後の特徴量の対数周波数軸方向の長さを表す．

表 5 各畳み込み層・転置畳み込み層のパラメータ

層名	C_{out}	カーネル	ストライド	パディング
First1x1	64	1×1	1×1	0×0
Down3x4	64	3×4	1×2	1×1
Conv1x1(s1x1, p0x0)	64	1×1	1×1	0×0
Conv1x5(s1x1, p0x2)	64	1×5	1×1	0×2
Conv3x5(s1x1, p1x2)	64	3×5	1×1	1×2
Conv4x5(s2x1, p1x2)	64	4×5	2×1	1×2
Conv4x5(s2x1, p2x2)	64	4×5	2×1	2×2
Deconv1x1(s1x1, p0x0)*	64	1×1	1×1	0×0
Deconv1x5(s1x1, p0x2)*	64	1×5	1×1	0×2
Deconv3x5(s1x1, p1x2)*	64	3×5	1×1	1×2
Deconv4x5(s2x1, p1x2)*	64	4×5	2×1	1×2
Deconv4x5(s2x1, p2x2)*	64	4×5	2×1	2×2
Up3x4*	64	3×4	1×2	1×1
Last1x1	1	1×1	1×1	0×0
ClassifierLast1x1	K	1×1	1×1	0×0

分類器は 4 層の畳み込み層と 1 層の Global Average Pooling 層 [9] から構成される．畳み込み層のカーネルサイズはすべて 1×1 である．4 層目の畳み込み層（ClassifierLast1x1）の出力チャネル数は分類するドメインの数 K である．Global Average Pooling 層では入力される特徴量について時間軸方向の平均が計算され，サイズ (K) のクラス分類が出力される．クラス分類は後述する目的関数に含まれる softmax 関数と合わせて，中間表現がどの音色の音楽音響信号からエンコーダにより変換されたものであるかの分類結果の確率値を表す．

以上が各ネットワークの構成である．また，エンコーダ，デコーダ中のすべての畳み込み層と転置畳み込み層の重みには Weight Normalization[10] を適用している．畳み込み層や転置畳み込み層またはその出力に適用可能な正規化手法として Batch Normalization[11]，Instance Normalization[12]，Weight Normalization がある．これらを用いた場合および正規化手法を用いない場合のそれぞれで学習の進行を比較したところ，Weight Normalization を用いた場合にのみ安定した学習の進行を確認したため，Weight Normalization を採用している．また，分類器中のすべての畳み込み層の重みには Spectral Normalization[13] を適用している．Spectral Normalization は敵対的生成ネットワーク（GAN）[14] の学習を安定化させるために提案された正規化手法である．Ganin らはドメイン敵対的ニューラルネットワークの学習においても既存の GAN の最適化手法を利用できるとしており [5]，提案手法では音色変換モデルの学習を安定化させるために分類器に Spectral Normalzation を適用している．

3.4 対数周波数対数振幅スペクトログラムから音楽音響信号への再合成

デコーダから出力された 4 つの対数周波数対数振幅スペクトログラム $Y_{\mathrm{logcq,norm}}[k,n]$ は次式により対数周波数振幅スペクトログラム $Y_{\mathrm{cq}}[k,n]$ に変換される．

$$Y_{\mathrm{cq}}[k,n] = X_{\max}[k] \exp\left((-\log\varepsilon)(Y_{\mathrm{logcq,norm}}[k,n] - 0.5)\right)$$

変換された対数周波数振幅スペクトログラムのそれぞれに Grifiin-Lim 法を適用し 4 つの音楽音響信号へ再合成する．本研究では Python の librosa パッケージ [15] に含まれる griffinlim_cqt 関数を用いて Griffin-Lim 法の計算を行った．griffinlim_cqt では Perraudin らの高速な Griffin-Lim アルゴリズム（FGLA）[16] が用いられている．FGLA により再合成された 4 つの音楽音響信号を入力の音楽音響信号と同じサンプリング周波数にリサンプリングし，それらの音楽音響信号を加算した結果を提案する音色変換システムの出力とする．

3.5 音色変換モデルの学習

提案手法の音色変換モデルは次の目的関数を確率的勾配降下法により最適化することでパラメータの学習を行う．

$$\mathcal{L} = \frac{1}{K}\sum_{k=1}^{K} \mathop{\mathbb{E}}_{x_k \sim X_k} \left[|D(E(x_k)) - x_k| - \lambda \mathcal{L}_{\mathrm{cls}}(C(E(x_k)), k) \right]$$

ただし，$k = 1, 2, \cdots, K$ は変換対象の音色（ドメイン）に対応する添字，K はドメインの数であり，E はエンコーダ，D_k は k 番目の音色用のデコーダ，C は分類器である．$\mathcal{L}_{\mathrm{cls}}(y,k)$ は softmax 関数を適用した y に対するドメイン k を正解クラスとするクロスエントロピー誤差である．λ は分類誤差の項に対する重みの定数である．エンコーダ E および各デコーダ $D_1, D_2, \cdots, D_K$ は目的関数 $\mathcal{L}$ を最小化するように学習し，分類器は $\mathcal{L}$ を最大化するように学習する．

4. 評価実験

評価実験では 2 つの異なる実験を行った．一つは，提案手法の音色変換性能を客観的な評価指標を用いて検証する実験である．もう一つは，Mor らの手法と提案手法の間の音色変換性能の差を主観評価から検証する実験である．客観評価実験では，音色変換前の音楽音響信号と音色変換後の正解の音楽音響信号のペアからなるパラレルなデータセットを用いた．主観評価実験では，音色変換後の正解の音楽音響信号をデータとして含まない非パラレルなデータセットを用いた．

4.1 パラレルデータによる客観評価実験

4.1.1 実験目的と設定

客観評価実験では，提案手法の音色変換性能を客観的な評価指標により検証する．データセットとして電子的な演奏情報の一つである MIDI データが楽曲ごとに記録されているものを使用する．MIDI データからは 2 つの異なる音色の音楽音響信号を合成する．合成された 2 つの音楽音響信号の一方を変換前の音色の音楽音響信号，もう一方を変換後の正解の音楽音響信号として，音色変換システムの入出力のパラレルデータを作成する．作成されたパラレルデータを用いて変換後の音楽音響信号と正解の音楽音響信号の誤差，変換前の音楽音響信号と正解の音楽音響信号の誤差をそれぞれ計算し，それら 2 つの誤差を比較する．これにより，変換後の音楽音響信号が変換前に比べてどれだけ正解の音楽音響信号に近づいたかを客観的に評価する．音楽音響信号間の誤差として 2 つの計算式を用いる．一つは次式により表される振幅スペクトログラムの周波数軸方向のユークリッド距離の時間平均である．

$$d_{\mathrm{STFT}}(x,t) = \frac{1}{T}\sum_{n=0}^{T-1} \sqrt{\sum_{k=0}^{b-1} (X_{\mathrm{STFT}}[k,n] - T_{\mathrm{STFT}}[k,n])^2}$$

ただし，$X_{\mathrm{STFT}}[k,n]$ および $T_{\mathrm{STFT}}[k,n]$ はそれぞれ音楽音響信号 $x[n]$, $t[n]$ の振幅スペクトログラム，b および T は振幅スペクトログラムの周波数ビン数とフレーム数である．もう一つの誤差の計算式は次式により表される対数振幅スペクトログラムの周波数軸方向のユークリッド距離の時間平均である．

$$d_{\mathrm{logSTFT}}(x,t) = \frac{1}{T}\sum_{n=0}^{T-1} \sqrt{\sum_{k=0}^{b-1} \left(\log \frac{X_{\mathrm{STFT}}[k,n] + \varepsilon}{T_{\mathrm{STFT}}[k,n] + \varepsilon} \right)^2}$$

ただし，ε は計算の発散を防ぐための微小な値である．それぞれの誤差について，データセットから抽出したテストデータ全体に対する誤差の平均を評価指標とする．

表 6 客観評価実験結果

	変換後-正解間誤差	変換前-正解間誤差
振幅誤差	6.27	7.55
対数振幅誤差	17.3	16.5

音色変換モデルの学習および評価実験に用いるデータセットとして MusicNet[17] の MIDI データ 330 曲を使用した．それぞれの MIDI データに対して Python パッケージの pretty_midi[18] を用いて音楽音響信号を合成した．合成のための音源には pretty_midi に付属するサウンドフォント（TimGM6mb.sf2）を使用し，各 MIDI データに対してアコースティックピアノ，マリンバ，アコースティックギター，フルートの音楽音響信号をそれぞれ合成した．合成した音楽音響信号のうち 281 曲分を学習データとして提案手法の音色変換モデルを学習した．学習時のイテレーション回数は 128360 回，バッチサイズは 4，目的関数における分類誤差の重みは $\lambda = 0.1$，最適化器は Adam[19]（ハイパーパラメータは $\beta_1 = 0.9, \beta_2 = 0.999$），学習率は 10^{-3}

である．データセットのうち学習に用いていない楽曲から4秒の区間を100個抽出し，評価指標の計算に用いるテストデータとして使用した．

4.1.2 結果

実験結果を**表 6**に示す．第2列は変換後の音楽音響信号と正解の音楽音響信号の間の誤差について異なる音色間でのすべての変換の平均を計算したものである．第3列は変換前の音楽音響信号と正解の音楽音響信号の間の誤差について異なる音色間でのすべての変換の平均を計算したものである．実験結果から，振幅スペクトログラム誤差については変換後の誤差が変換前の誤差よりも小さいことが分かる．また，対数振幅スペクトログラム誤差については変換後の誤差が変換前の誤差よりも大きいことが分かる．

4.1.3 考察

対数振幅スペクトログラムは振幅スペクトログラムに比べて高音域の周波数成分を表しやすいという特性がある．楽音は基本周波数成分とその整数倍の周波数の成分から構成され，基本周波数成分は楽音の音高に，すべての周波数成分は楽音の音色に強く影響する．これらのことから，提案手法の音色特徴の特に高音域における変換性能はいま一つであるが，変換前の音楽音響信号に存在する音高は維持されていると考えられる．

4.2 非パラレルデータによる主観評価実験

4.2.1 実験目的と設定

主観評価実験では，音色の変換性能と音高の維持性能のそれぞれを提案手法と先行研究のMorらの手法の間で比較する．比較対象であるMorらの手法の変換モデルについて，スタイル変換モデルとして学習済みのものを実験に使用した．それに従い，本研究の提案手法の音色変換モデルについても，先行研究と同一の条件でスタイル変換モデルとして学習した．データセットとしてMusicNet[17]の録音データから203曲を使用した．**表 7**に変換対象とする6つのスタイルとデータセットに含まれる楽曲の例を示す．データセットのうち160曲約60時間分を学習データとした．学習時のイテレーション回数は101400回，その他の学習パラメータは客観評価実験と同様に設定した．データセットのうち学習データとして使用していない録音データをテストデータとした．

主観評価のために12名の被験者を対象に各手法の変換例に関するテストを2つ実施した．一つ目のテストは音色の類似性に関するABXテストである．このテストでは設問ごとに3つの音楽音響信号が与えられる．うち2つは異なる2つのスタイルの音楽音響信号であり，それぞれ「A」または「B」のラベルが付けられている．もう一つの音楽音響信号は，「A」と「B」の2つのスタイルの一方を変換元のスタイル，もう一方を変換先のスタイルとしたスタイル変換を提案手法またはMorらの手法のいずれかの手法により行った後の音楽音響信号である．この音楽音響信号には「X」のラベルが付けられている．被験者には「X」の音楽音響信号の音色について「A」の音楽音響信号と「B」の音楽音響信号のうちより音色が近いほうを選択してもらった．このとき，「X」の変換先のスタイルの音楽音響信号を選択した場合が正解となる．これを，使用した手法とスタイルが被験者に分からない形で，2つの手法について異なるスタイルの組み合わせを1問ずつ，計60問設定した．収集された選択結果から正解率を計算し評価指標とする．

もう一つのテストは音高の同一性に関する5段階評価のテストである．このテストでは設問ごとに異なるスタイル間での変換前後の音楽音響信号が与えられる．このとき，2つの音楽音響信号のどちらが変換後のものかを音楽音響信号以外の情報から判別されないように問題を設定した．被験者は2つの音楽音響信号を聴き比べた上で，次の5段階の評価のうち一つを選択する．

5. すべての音の高さが同じであると感じる
4. すべての音の高さが同じではないが，同じ箇所のほうが多いと感じる
3. 音の高さが同じ箇所と異なる箇所が同程度あると感じる
2. すべての音の高さが異なってはいないが，異なる箇所のほうが多いと感じる
1. すべての音の高さが異なると感じる

これを，使用した手法とスタイルが被験者に分からない形で，2つの手法について異なるスタイルの組み合わせを1問ずつ，計60問設定した．収集された評価結果から手法ごとに平均オピニオン評点（MOS）を計算し評価指標とする．MOSは5段階評価のそれぞれに割り当てられた評価値（$5, 4, 3, 2, 1$のいずれか）の平均である．

4.2.2 結果

まず，音色の類似性に関するABXテストの実験結果を**表 8**に示す．スタイル変換後の音楽音響信号について，変換元のスタイルの音楽音響信号よりも変換先のスタイルの音楽音響信号のほうが音色が近いと回答した割合はMorらの手法では72.8%であり，提案手法では46.9%であった．

続いて，音高の同一性に関する5段階評価のテストの実験結果を**表 9**に示す．5段階評価の平均値はMorらの手法では3.25であり，提案手法では3.58であった．

4.2.3 考察

音色の類似性に関する結果から，提案手法の音色変換性能はMorらの手法に比べて大きく劣ることが分かる．提案手法における正解率はおよそ50%であり，提案手法を用いる場合，変換元の音色とも変換先の音色とも言い難い音色の音楽音響信号に変換されることが分かる．また，音高の同一性に関する結果から，提案手法は先行研究に比べて変換前の音楽音響信号に存在する音高を維持できていることが分かる．以上の2つのテストから，客観評価実験からも

表 7 主観評価実験の変換対象のスタイル一覧

スタイル	楽曲の例
チェロソロ（バッハ）	無伴奏チェロ組曲 第 4 番 変ホ長調
ピアノソロ（バッハ）	平均律クラヴィーア曲集 第 1 巻 第 1 番 ハ長調
伴奏付きヴァイオリン（ベートーヴェン）	ヴァイオリンソナタ 第 9 番 イ長調
ピアノソロ（ベートーヴェン）	ピアノソナタ 第 8 番 ハ短調
弦楽四重奏（ベートーヴェン）	弦楽四重奏曲 第 7 番 ヘ長調
木管五重奏（カンビーニ）	木管五重奏曲 第 1 番 変ロ長調

表 8 音色類似性の実験結果

手法	正解率（%）
Mor らの手法	72.8
提案手法	46.9

表 9 音高同一性の実験結果

手法	MOS
Mor らの手法	3.25
提案手法	3.58

示されたように，提案手法の音色特徴の変換性能はいま一つである．しかしながら，変換前の音楽音響信号に存在する音高を変換後の音楽音響信号で維持することができており，Mor らの手法の問題点である音高の消失を抑制する効果が提案手法にはあることが確認できる．提案手法における音高の維持性を保ちながら Mor らの手法と同じ水準まで音色変換の性能を向上させるために，音色変換モデルの構成の見直しや補助情報の利用の検討を今後の課題とする．

5. おわりに

本研究ではドメイン敵対的ニューラルネットワークの学習手法を用いた対数周波数対数振幅スペクトログラム上の音色変換モデルによる音色変換システムを提案した．客観評価実験および主観評価実験のそれぞれから，提案手法は先行研究の Mor らの手法に比べて音色変換の性能は劣るが，本研究で目的としていた変換前の音楽音響信号に存在する音高の消失の抑制には寄与していることを確認した．

謝辞 本研究は JSPS 科研費 JP20K12126 の助成を受けたものです．

参考文献

[1] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D. and Simonyan, K.: Neural audio synthesis of musical notes with wavenet autoencoders, *International Conference on Machine Learning*, PMLR, pp. 1068–1077 (2017).

[2] Mor, N., Wolf, L., Polyak, A. and Taigman, Y.: A Universal Music Translation Network (2018).

[3] Brown, J. C.: Calculation of a constant Q spectral transform, *The Journal of the Acoustical Society of America*, Vol. 89, No. 1, pp. 425–434 (1991).

[4] Griffin, D. and Lim, J.: Signal estimation from modified short-time Fourier transform, *IEEE Transactions on acoustics, speech, and signal processing*, Vol. 32, No. 2, pp. 236–243 (1984).

[5] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. and Lempitsky, V.: Domain-adversarial training of neural networks, *The journal of machine learning research*, Vol. 17, No. 1, pp. 2096–2030 (2016).

[6] Young, R. W.: Terminology for logarithmic frequency units, *The Journal of the Acoustical Society of America*, Vol. 11, No. 1, pp. 134–139 (1939).

[7] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[8] Zeiler, M. D., Krishnan, D., Taylor, G. W. and Fergus, R.: Deconvolutional networks, *2010 IEEE Computer Society Conference on computer vision and pattern recognition*, IEEE, pp. 2528–2535 (2010).

[9] Lin, M., Chen, Q. and Yan, S.: Network In Network (2014).

[10] Salimans, T. and Kingma, D. P.: Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks (2016).

[11] Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift (2015).

[12] Ulyanov, D., Vedaldi, A. and Lempitsky, V.: Instance Normalization: The Missing Ingredient for Fast Stylization (2017).

[13] Miyato, T., Kataoka, T., Koyama, M. and Yoshida, Y.: Spectral Normalization for Generative Adversarial Networks (2018).

[14] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative Adversarial Networks (2014).

[15] McFee, B., Metsai, A., McVicar, M., Balke, S., Thomé, C., Raffel, C., Zalkow, F., Malek, A., Dana, Lee, K., Nieto, O., Ellis, D., Mason, J., Battenberg, E., Seyfarth, S., Yamamoto, R., viktorandreevichmorozov, Choi, K., Moore, J., Bittner, R., Hidaka, S., Wei, Z., nullmightybofo, Hereñú, D., Stöter, F.-R., Friesch, P., Weiss, A., Vollrath, M., Kim, T. and Thassilo: librosa/librosa: 0.8.1rc2 (2021).

[16] Perraudin, N., Balazs, P. and Søndergaard, P. L.: A fast Griffin-Lim algorithm, *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE, pp. 1–4 (2013).

[17] Thickstun, J., Harchaoui, Z. and Kakade, S. M.: MusicNet (2016).

[18] Raffel, C. and Ellis, D. P. W.: Intuitive Analysis, Creation and Manipulation of MIDI Data with pretty_midi (2014).

[19] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization (2017).