Vol.2019-MUS-125 No.3
2019/11/19

Normalizing Flowを用いたシンセサイザー制御学習

増田 尚建[1,a)]　フィリップ　エスリン[2]　伊庭　斉志[1]

概要：シンセサイザーは現代の音楽制作において欠かせないツールとなっている．一方で，シンセサイザーはパラメータが多く複雑なため，求めているような音を作ることは困難である．本研究ではシンセサイザー制御の問題を定式化し，音の潜在空間とパラメータとの対応を Normalizing flow を用いて学習する．変分オートエンコーダを用いて得られた音の潜在表現を用いることで，パラメータの数が多い場合にも正確なパラメータ推定を行うことができる．そして，音の潜在空間のもつれを解くことで意味情報に結びついた次元によるシンセサイザーの操作を可能にする．

1. はじめに

近年，ソフトシンセサイザーの普及により，音楽制作においてシンセサイザーがより頻繁に用いられるようになっている．ソフトシンセサイザーは容易に DAW（Digital Audio Workstation）ソフトでプラグインとして用いることができ，アナログシンセサイザーよりも安価で手に入れやすい．本研究でもソフトシンセサイザーを対象として実験を行う．

シンセサイザーはパラメータを調整することで 1 つのシンセサイザーから様々な音色の音を出力することができる．よって，既存の楽器を模倣したような音に限らず新しい合成音を創り出すこともできる．一方で，シンセサイザーのパラメータを適切に調整し，求めているような音を出力させることは困難である．これはパラメータの多さと，パラメータと出力の音との関係が非線形であることに起因する．

シンセサイザーの操作は難しいため，実際の音楽制作においてプリセットを用いることが多い．プリセットとは特定の音を出すようなパラメータの設定ファイルであり，有志により作られたものがインターネットでダウンロードできるほか，有償配布もなされている．ただし，多くのファイルの中から自分の求めているような音を出すプリセットを見つけ出すことは困難である．また，プリセットに求める音がない場合や新しい音が作りたい場合も多い．

よって，シンセサイザーによる音の合成をより簡単で直感的にするようなシステムの需要は大きい．そのようなシステムを作るためにはシンセサイザーのパラメータと音との関係をモデリングする必要がある．この関係を学習することを本稿ではシンセサイザー制御学習と呼ぶ．

シンセサイザーによる音の合成を支援するシステムもいくつか提案されている．最も単純な手法としてユーザに作りたい音の例をクエリとして入力させ，その音をシンセサイザーで出力するようなパラメータを推定することが考えられる．本稿ではこのタスクをパラメータ推定と呼ぶ．Yee-king ら [1] は bi-directional LSTM を用い，クエリ音の MFCCs（Mel-Frequency Cepstrum Coefficients）からシンセサイザーのパラメータを推定した．このようなシステムはユーザが自身の求めている音を知っていて，それをクエリとして提供できることを前提としているが，実際にそのような条件を満たすユースケースは少ない．Cartwright ら [2] は求めている音をユーザが声で模倣し，その音声をクエリとした検索によりシンセサイザーの制御を支援するシステムを提案した．声による音の模倣は不完全なため，適合性フィードバックを用い，ユーザが対話的に検索結果を改善できるようにした．

音の合成を簡単にするためのもう一つのアプローチとして，シンセサイザーに「マクロコントロール」を導入することが考えられる．マクロコントロールとは，シンセサイザーのパラメータにマッピングされたツマミであり，一つのツマミで複数のパラメータを同時に調整することができる（図 1 参照）．多くの場合マクロコントロールは音のある特徴（例：明るさ，激しさ）でラベル付けされている．これにより，パラメータが多すぎるという問題と，パラメータが音にどのような影響を与えるか分からないという問題を解決できる．しかし，このようなマクロコントロールは

[1] 東京大学　大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
[2] フランス国立音響音楽研究所
Institut de Recherche et Coordination Acoustique/Musique
[a)] masuda@iba.t.u-tokyo.jp

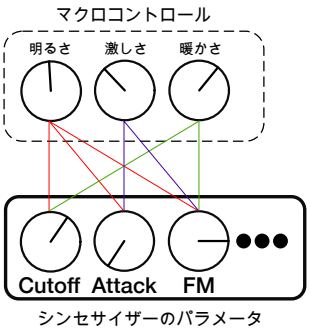

図 1 シンセサイザーにおけるマクロコントロールのイメージ.

人間が設定しなければならない．また，単純なマッピングによって設計されたマクロコントロールがある音の特徴の変化に対応するのは，パラメータ空間においても局所的な空間に限る．よって，一つ一つのプリセットごとにマクロコントロールを設計しなければならない．

本研究では，normalizing flow[3] を用い，シンセサイザーのパラメータと合成音との関係を学習するモデルを提案する．音をそのまま用いず，Variational Auto-Encoder によって音の潜在空間をモデリングし，音の潜在空間とパラメータとの関係を学習することで，より音の構造を保ったパラメータ推定が行える．潜在空間をさらに NF で disentangle する（もつれを解く）ことで，意味情報に結びついたマクロコントロールを獲得する．

2. 背景

2.1 Variational Auto-encoder

Variational Auto-encoder[4]（VAE）とは生成モデルを学習するための手法である．生成モデルではデータがどのような分布から生じているかを学習する．データ x が生成的要因（潜在変数）z を持つ確率分布 $p(x)$ からサンプリングされたとする．$p(x)$ についての最尤法により，以下の式を最大化すればよい．

$$p(x) = \int p(x|z)p(z) \tag{1}$$

これを直接求めることはできないため，z の事後確率を $q(z|x)$ と近似する．$p(z|x)$ と $q(z|x)$ の Kullback-Leibler（KL）ダイバージェンスは下式の通りとなる．

$$\begin{aligned} &D_{KL}[p(z|x)|q(z|x)] \\ &\quad = E_{z\sim q}[\log q(x|z)] - \log q(z|x) - \log p(z|x) \end{aligned} \tag{2}$$

これより，

$$\begin{aligned} &\log p(x) - D_{KL}[q(z|x)|p(z|x)] \\ &\quad = E_{z\sim q}[\log p(x|z)] - D_{KL}[q(z|x)|p(z)] \end{aligned} \tag{3}$$

上式の左辺はモデリングしたい分布と，十分に表現力の高いエンコーダ q をとれば 0 に近づく誤差項の和である．

よって目的関数は

$$\mathcal{L}(\theta; X) = E_{z\sim q}[\log p(x|z)] - D_{KL}[q(z|x)|p(z)] \tag{4}$$

$E_{z\sim q}[\log p(x|z)]$ はデコーダによる再構成誤差である．$D_{KL}[q(z|x)|p(z)]$ は reparameterization trick を用いて計算できる．エンコーダのネットワークが正規分布 $q(z|x)$ の分散と平均を出力する．$p(z)$ は平均が 0，分散が 1 の正規分布とする．

もつれのほどかれた表現

学習した表現のそれぞれの次元が独立であり，それぞれ特定の生成的要因に対応している場合，表現の「もつれがほどけている」という． Beta-VAE[5] などの手法は，もつれのほどかれた表現を教師なしで学習する．Beta-VAE では VAE の目的関数の KL divergence 項に 1 以上の重み β をかけることで，もつれのほどかれた表現を学習させる．教師なし学習の手法では，単純なデータについてしかもつれのほどかれた表現を学習できない．Fader Networks[6] などの教師ありの手法はラベルに基づいて潜在変数のもつれを解くことに成功している．

直感的に解釈できる潜在変数を獲得するようにもつれを解けば，人が出力を制御する生成モデルを実現することができる．NSynth[7] は WaveNet ベースのオートエンコーダを用いて楽器音の潜在空間を学習した．このモデルではエンコーダに音を入力し得られる潜在表現について内挿を取ることで，２つの楽器を掛け合わせたような新しい音が得られる．しかし，潜在空間そのものはもつれているため，より直感的な操作は不可能である．Bitton ら [8] は Wasserstein Auto-Encoder（WAE）[9] ベースの Fader Network を用い，音程・音色・奏法などの特徴を明示的に制御できる生成モデルを提案した．

2.2 Normalizing Flow

Normalizing flow は単純な分布をより複雑な分布へと変換することで，サンプリングのコストを抑えつつも表現力の高い確率分布を得るための手法である．z_{t-1} に対し可逆な写像 f_t を行い $z_t = f_t(z_{t-1})$ を得る．確率密度の変数変換の法則により下式が得られる．

$$q(z_t) = q(z_{t-1}) \left| \det \frac{\partial \mathrm{f_t}^{-1}}{\partial \mathrm{z_{t-1}}} \right| = q(z_{t-1}) \left| \det \frac{\partial \mathrm{f_t}}{\partial \mathrm{z_{t-1}}} \right|^{-1} \tag{5}$$

最初の分布 z_0 に対しこのような変数変換を繰り返し行うと z_k の確率分布は下式の通りとなる．

$$\log q(z_k) = \log q(z_0) - \sum_{t=1}^{T} \left| \det \frac{\partial \mathrm{f_t}}{\partial \mathrm{z_{t-1}}} \right| \tag{6}$$

この変数変換の連鎖を normalizing flow と呼ぶ．これを計算するには，ヤコビアンが効率的に計算できるような変換 f を考える必要がある．Inverse Autoregressive Flow（IAF）[10] や Masked Autoregressive Flow（MAF）[11] など様々な種類の flow が提案されている．

3. 提案手法

3.1 シンセサイザー制御学習

あるシンセサイザーから出力される音の群を $D = \{\mathbf{x_i}\}$，その潜在変数を $\mathbf{z}$ としたとき，音の潜在変数モデルは $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ と表せる．

パラメータ $\mathbf{v}$ を与えられたシンセサイザーのモデルは下式のように表せる．

$$p(\mathbf{x}, \mathbf{v}, \mathbf{z}) = p(\mathbf{x}|\mathbf{v}, \mathbf{z})p(\mathbf{v}|\mathbf{z})p(\mathbf{z}) \tag{7}$$

シンセサイザーはパラメータを入力とし，音を出力とする関数と考えられる．シンセサイザー f_s によって合成音 $D_s \in D$ が出力されるとき，

$$\mathbf{x} = f_s(\mathbf{v}); \mathbf{x} \in D_s \tag{8}$$

シンセサイザーから出力された音は，正しいパラメータを見つけられれば同じシンセサイザーで再現可能なはずである．それ以外の音については，

$$\mathbf{x} = f_s(\mathbf{v}) + \epsilon; \mathbf{x} \notin D_s \tag{9}$$

ただし，ϵ は $\mathbf{x}$ の再現誤差である．

$\mathbf{z}$ が与えられたとき，$\mathbf{x}$ と $\mathbf{v}$ は独立であるとする．

$$\log p(\mathbf{x}, \mathbf{v}, \mathbf{z}) = \log\{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\} + \log \mathbf{p}(\mathbf{v}|\mathbf{z}) \tag{10}$$

(10) 式右辺の第 1 項は VAE により求められる．本研究ではパラメータと潜在変数との関係を指す $p(v|z)$ を normalizing flow を用いて求める．

潜在空間 $\mathbf{z}$ では，似ている音が近くなるような分布となっている．よって，潜在空間 $\mathbf{z}$ の各次元をツマミとみなし移動させると，音がだんだんと変わるとともに，その音を出力するようなパラメータも求まる．よって，これをマクロコントロールとして用いることが考えられる．一方で，どの次元がどのような感性的意味合いを持つのかは明示されていない．

Normalizing flow は可逆な変換であるため，既存のプリセットについて音の潜在変数を得ることができる．あるプリセットについて音の潜在空間によって操作することで，プリセットの自然な調整や 2 つのプリセットの合成のような新しい操作方法が得られる．

3.2 マクロコントロール学習

本研究ではシンセサイザーのプリセットにラベル付けされた特徴タグを用いて潜在空間のもつれを解くことにより，意味的マクロコントロールを実現する．特徴タグ $\mathbf{t}$ を考慮すると，

$$\begin{aligned} p(\mathbf{x}, \mathbf{z}, \mathbf{v}, \mathbf{t}) &= p(\mathbf{x}|\mathbf{v}, \mathbf{t}, \mathbf{z})p(\mathbf{v}|\mathbf{z}, \mathbf{t})p(\mathbf{t}|\mathbf{z})p(\mathbf{z}) \\ &= p(\mathbf{x}|\mathbf{t}, \mathbf{z})p(\mathbf{v}|\mathbf{z}, \mathbf{t})p(\mathbf{t}|\mathbf{z})p(\mathbf{z}) \end{aligned} \tag{11}$$

生成モデル $p_\theta(\mathbf{x}|\mathbf{t}, \mathbf{z})$ について考える．カテゴリカル変数 $\mathbf{t}$ の事前分布は $p(\mathbf{t}) = Cat(\mathbf{t}|\pi)$ $p(\pi)$ とおく．認識モデルは $q_\phi(\mathbf{z}, \mathbf{t}|\mathbf{x})$ とおく．さらに，$q_\phi(\mathbf{z}, \mathbf{t}|\mathbf{x}) = \mathbf{q}_\phi(\mathbf{z}|\mathbf{x})\mathbf{q}_\phi(\mathbf{t}|\mathbf{x})$ と仮定する．

ただし，特徴タグがプリセットデータから欠けていることも多い．この問題を解決するため，Kingma ら [12] の半教師あり生成モデルを参考にする．

$\mathbf{t}$ が未知の場合，$\mathbf{z}$ と同様に $\mathbf{t}$ も潜在変数と考える．

$$\begin{aligned} L_u = -\mathbb{E}[\log p_\theta(\mathbf{x}|\mathbf{t}, \mathbf{z}) + \log p_\theta(\mathbf{t}) + p_\theta(\mathbf{z})] \\ - \mathbb{E}[\log q_\phi(\mathbf{t}, \mathbf{z}|\mathbf{x})] \end{aligned}$$

$\mathbf{t}$ が既知の場合，「明るい」と「暗い」のように対をなす特徴タグの組について，それぞれ目標分布 $p(x_{\mathbf{t}_-}) \sim \mathcal{N}(-\mu_*, \sigma_-)$ と $p(x_{\mathbf{t}_+}) \sim \mathcal{N}(+\mu_*, \sigma_+)$ を定義する．$D_{KL}[q_\phi(z_{\mathbf{t}_*}|\mathbf{x})||p(z_{\mathbf{t}_*})]$ を最小化することで，既知のタグ t_* を目標分布に近づける．そのため，$q_\phi(z_{\mathbf{t}_*}|\mathbf{x})$ を潜在変数 z_* に normalizing flow f_k を加えてモデリングする．

もつれを解くための normalizing flow の目的関数は

$$\begin{aligned} L_o &= D_{KL}[q_\phi(z_{\mathbf{t}_*}|\mathbf{x})||p(z_{\mathbf{t}_*})] \\ &= \mathbb{E}\left[\log p(z) + \sum_{i=1}^{k} \log|\det \frac{\partial f_i}{\partial z_{i-1}}| - \log p(z_{t_*})\right] \end{aligned}$$

モデル全体の目的関数は $L = L_u + L_o$ となる．そして，マクロコントロールを含むモデル全体の概要図は図 2 のようになる．

4. 実験概要

4.1 データセット

あるシンセサイザーのパラメータと出力音の関係をモデルに学習させるためには，シンセサイザーのパラメータデータとその出力音を対応付けたデータセットが必要となる．Yee-king ら [1] はランダムにパラメータを生成したが，パラメータの数が大きくなると，パラメータ空間全体を十分に覆うようにサンプリングすることは困難となる．さらに，ランダムでパラメータを設定すると殆どの場合，音楽には使えないような無意味な音が出力される．よって，本研究ではプリセットをインターネットから集め，シンセサイザーでそれらを再生・録音することでデータセットを作成した．

シンセサイザーは u-he 社の *Diva*[*1] を対象とした．広く用いられているシンセサイザーであるためプリセットが多く公開されている上，プリセットがパースできるような形式になっている．減算式のシンセサイザーで比較的単純なシグナルフローを持ち，WaveTable 機能などモデリングの困難なパラメータを持たない．

Diva には 200 以上のパラメータがあるが，本研究ではその中でも重要な 16・32 個についてのみ扱う．パラメータ

[*1] https://u-he.com/products/diva/

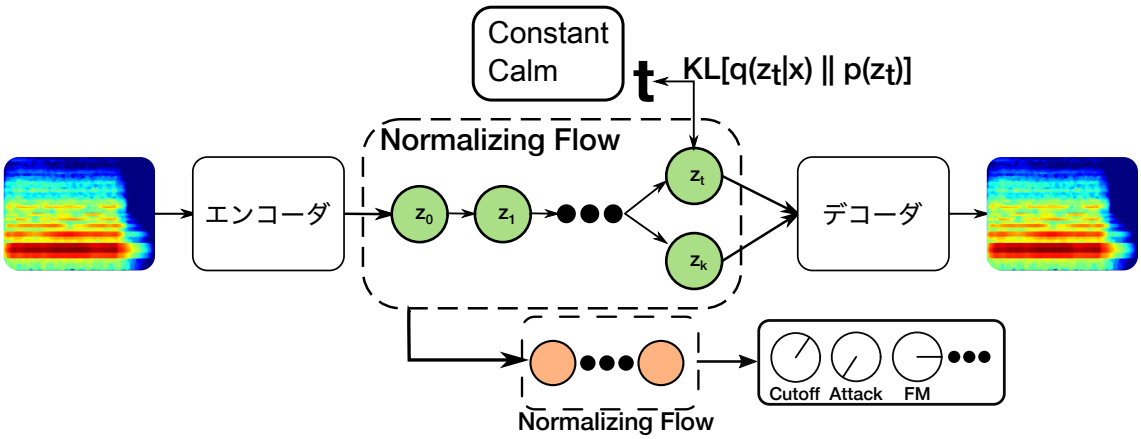

図 2 提案手法の概要図.

の数を制限しても，シンセサイザーの出力できる音は十分に幅広い．扱わないパラメータはシンセサイザーのデフォルトの値のまま固定した．本来のプリセットに対応する出力音とは若干異なるが，大まかな特徴は十分に保っている．約 11,000 個のプリセットに対し，それぞれ同じピッチとベロシティで再生・録音した．ノートは 3.5 秒保持し，リリースも捉えるため 4 秒間録音した．VST 形式のシンセサイザーを自動で再生・録音するため，プリセットを扱えるように改変した RenderMan ライブラリ[*2]を用いた．

プリセットに含まれた特徴タグも意味的マクロコントロールを学習するのに用いる．特徴タグは「明るい・暗い」や「動的・静的」のように対になるような形容詞の組としてメーカーにより定義されており，そのどちらかがプリセットの製作者によりプリセットにタグ付けされている．形容詞対のどちらもタグ付けされていない場合もある．

4.2 モデル詳細

ベースライン

提案手法の有用性を確認するため，メル周波数スペクトログラムを入力とし，パラメータを推測するニューラルネットを 3 つ実装した．多層パーセプトロン（MLP）モデルは 5 層でそれぞれ 2048 ユニット，活性化関数は Exponential Linear Unit（ELU）とした．畳込みニューラルネット（CNN）モデルでは 128 チャンネルでカーネルサイズが 7，stride を 2 とし，Batch Normalization も用いた畳み込み層を 5 層，その後に 3 層の MLP を用いた．さらに，CNN と同じ設定の Residual Network（ResCNN）も実装した．

提案手法

3 つのオートエンコーダベースのモデルを用い，ベースラインと同様のメル周波数スペクトログラムについて学習させる．エンコーダとデコーダではベースラインと同様に CNN を用いたが，パラメータの数が全体で同じになるよう，エンコーダとデコーダのチャンネル数とユニット数を半分にした．決定論的なオートエンコーダ（AE），VAE，そして Wasserstein Auto-Encoder（WAE）を実装した．WAE では，VAE の KL Divergence の項を maximum mean discrepancy loss に置き換える．VAE の潜在変数に normalizing flow を加えたモデル（VAE_{flow}）を実装した．flow には Inverse Autoregressive Flow（IAF）を用いた．

以上のモデルでは，潜在変数からパラメータへの回帰を 2 層の MLP で行う．更に，VAE_{flow} については IAF を用いて潜在変数からパラメータへの回帰を行うモデル（$Flow_{reg}$）を実装する．そしてこのモデルについて特徴タグを用いて潜在変数のもつれを解くモデル（$Flow_{dis}$）を実装する．

訓練方法

全てのモデルについて ADAM を用い 500 エポック訓練した．最も複雑なモデルでも GPU に Titan Xp を搭載した PC で 5 時間で学習が終了する．

5. 実験結果・考察

5.1 パラメータ推定

パラメータ 16 個の場合，**パラメータ 32 個**の場合でシンセサイザーの合成音を再現するパラメータ推定を行った．そして，パラメータ 32 個のモデルで同じシンセサイザーで作ったものではない音（**シンセサイザー以外**，他のシンセサイザー・弦楽器などで出した音や音声を使用）についてもモデルに入力し，パラメータを推定した．パラメータ推定の結果は表 1 の通りである．元々シンセサイザーの出力音であるデータに対しては，その真のパラメータと推定のパラメータの平均二乗誤差を表示する．全てのパラメータは 0 から 1 の範囲をとる．そして，出力音の正確さの指標として，スペクトログラムに対する平均二乗誤差と Spectral Convergence を用いた．

[*2] https://github.com/fedden/RenderMan

	パラメータ 16 個			パラメータ 32 個			シンセサイザー以外	
	パラメータ	出力音		パラメータ	Audio		Audio	
	MSE_n	SC	MSE	MSE_n	SC	MSE	SC	MSE
MLP	0.236±.44	6.226±.13	9.548±3.1	0.218±.46	13.51±3.1	36.48±11.9	2.348±2.1	37.99±7.8
CNN	0.171±.45	1.372±.29	6.329±1.9	0.159±.46	19.18±4.7	33.40±9.4	2.311±2.2	29.22±8.2
$ResNet$	0.191±.43	1.004±.35	6.422±1.9	0.196±.49	10.37±1.8	31.13±9.8	2.322±1.6	31.07±9.5
AE	0.181±.40	0.893±.13	5.557±1.7	0.169±.40	5.566±1.2	17.71±6.9	1.225±2.2	27.37±7.2
VAE	0.182±.32	0.810±.03	4.901±1.4	0.153±.34	5.519±1.4	16.85±6.1	1.237±1.3	27.06±7.1
WAE	**0.159±.37**	0.787±.05	4.979±1.5	**0.147±.33**	3.967±.88	16.64±6.2	1.194±1.5	26.10±6.4
VAE_{flow}	0.199±.32	0.838±.02	4.975±1.4	0.164±.34	1.418±.23	17.74±6.8	1.193±1.8	27.03±6.4
$Flow_{reg}$	0.197±.31	**0.752±.05**	**4.409±1.6**	0.193±.32	**0.911±1.4**	**16.61±7.4**	**1.101±1.2**	**26.07±7.7**
$Flow_{dis.}$	0.199±.31	0.831±.04	5.103±2.1	0.197±.42	1.481±1.8	17.12±7.9	1.209±1.4	26.77±7.3

表 1 ベースライン・オートエンコーダモデル・提案手法のパラメータ推定結果の比較．パラメータの平均二乗誤差（MSE），出力音の Spectral Convergence（SC）・平均二乗誤差について平均と標準偏差を表記した．

パラメータ 16 個の場合，ベースラインも他のモデルと同程度に正確なパラメータ推定ができている．しかし，オートエンコーダモデル・提案手法に対して出力音についての指標が著しく劣っている．ベースラインはパラメータ空間上は近いが，出力音は大きく異なるようなパラメータを推定してしまっている．シンセサイザーではパラメータ全体の距離が近くても，一つの重要なパラメータがずれることで音が大きく変わってしまう．一方，オートエンコーダで音の潜在空間を学習することで，音についてより整った表現からのパラメータへの回帰を行うため，オートエンコーダを用いたモデルでは出力音についても正確なパラメータ推定が可能になったと考えられる．

パラメータ 32 個の場合，シンセサイザーの出力音の幅も大きく広がったためか，ベースラインの出力音の結果は著しく悪化した．パラメータ 16 個・32 個の両方で WAE モデルがパラメータの MSE については最も良い結果を示したが，出力音に対する指標では SC において提案手法が上回った．Normalizing flow によりパラメータへのマッピングを学習する提案手法は推定の正確さだけでなく，3 節で述べた通り可逆な変換によりもたらされる新しい操作方法が目的である．

マクロコントロールを学習するため潜在変数のもつれを解く目的関数を加えたモデルでは若干成績が悪化した．目標分布である特徴が比較的単純であるため，もつれを解かずに潜在変数により多くの情報量を詰め込めるモデルのほうが有利であるからだと考えられる．

パラメータ推定の結果の一例を図 3 に示す．パラメータ 16 個の場合には音が単純なためか，どちらも正確にパラメータ推定ができている．一方で，パラメータ 32 個の場合にはベースラインが全く音の時間的変化を再現できていないため，出力音が大きく異なる．これは時間的変化に影響するシンセサイザーのパラメータの重要性を捉えられていないためだと考えられる．提案手法はシンセサイザー以外の音に対しても近い音を出力した．

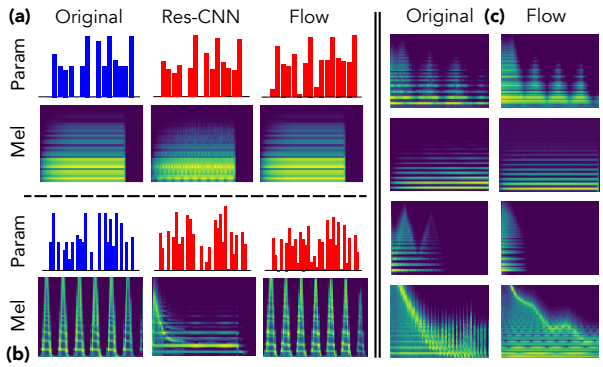

図 3 プリセットの出力音に対するパラメータ推定の結果．パラメータの値を棒グラフとして表示．(a) はパラメータ 16 個,，(b) はパラメータ 32 個，(c) はシンセサイザー以外の音に対するパラメータ推定．

5.2 マクロコントロール学習

マクロコントロール学習の結果は図 4 の通りである．もつれの解かれた次元における移動は，パラメータと非線形ながらもなめらかな関係を示している．スペクトログラムからもわかるように，どちらの次元も直感に即したような音の変化を見せている．「Calm/Aggressive」の次元で Aggressive の方に移動すると音がより激しくなっている．また，「Constant/Moving」の次元では Moving の方に移動するほど音に強弱の波が生じている．これらの次元を操作方法として用いることは十分に考えられる．ただし，音に対する感性は個人差も大きく，より正確に有用性を図るためにはユーザスタディが必要となる．

本手法で獲得できる意味的次元はプリセットに付けられた特徴タグに限られている．一方で，プリセットについてこのような意味的情報がラベル付けされていることは少ない．音に対する感性の個人差とデータの不足を解決する方法として，ユーザのフィードバックを考慮しつつもつれを解くシステムが考えられる．

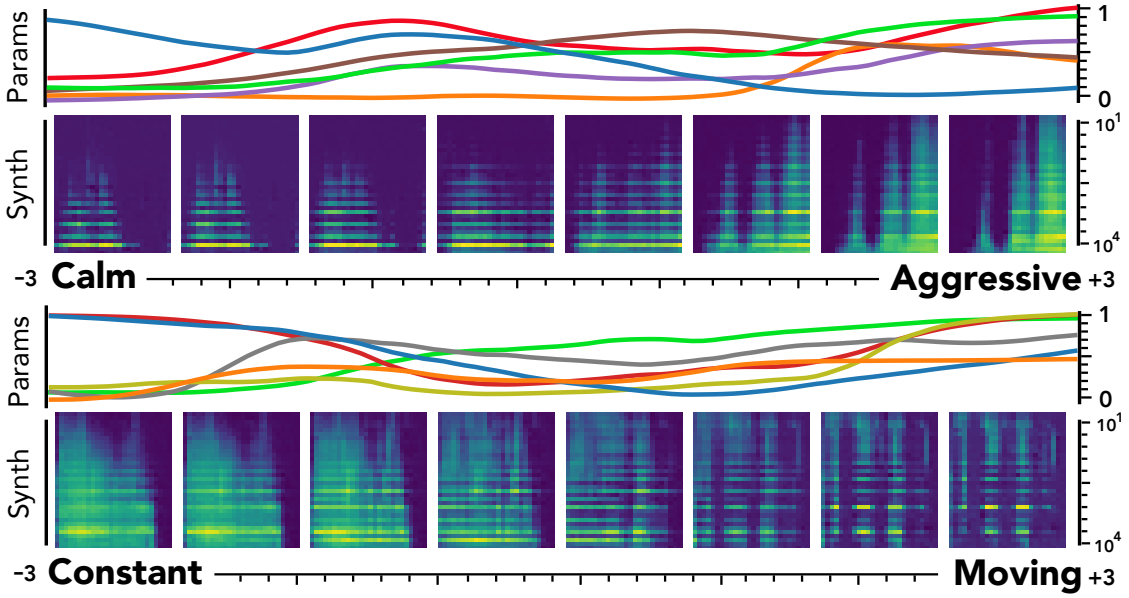

図 4 マクロコントロール学習の結果．もつれの解かれた次元 2 つのマクロコントロールとしての結果を示す．2 つの次元を動かしたときのパラメータの変化と，出力音のスペクトログラムを表示．

6. おわりに

本研究ではシンセサイザー制御学習，すなわちシンセサイザーのパラメータと出力音との関係を学習する問題を定式化した．そして，variational auto-encoder を用いシンセサイザーの合成音について潜在変数を獲得し，normalizing flow を用いこの潜在空間とパラメータ空間との関係を学習した．さらに，潜在変数に normalizing flow を用いもつれを解くことで，シンセサイザーについて意味情報に基づいた次元を獲得し，シンセサイザーの直感的な操作方法としてこれを用いることを提案した．

参考文献

[1] Yee-King, M. J., Fedden, L. and D'Inverno, M.: Automatic Programming of VST Sound Synthesizers Using Deep Networks and Other Techniques, *IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol. 2, No. 2, pp. 150–159 (2018).

[2] Cartwright, M. and Pardo, B.: SynthAssist: Querying an Audio Synthesizer by Vocal Imitation, *The International Conference on New Interfaces For Musical Expression (NIME)*, pp. 363–366 (2014).

[3] Rezende, D. J. and Mohamed, S.: Variational Inference with Normalizing Flows, *32nd International Conference on Machine Learning* (2015).

[4] Kingma, D. P. and Welling, M.: Auto-Encoding Variational Bayes, *International Conference on Learning Representations* (2014).

[5] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S. and Lerchner, A.: beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, *International Conference on Learning Representations* (2017).

[6] Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L. and Ranzato, M.: Fader Networks: Manipulating Images by Sliding Attributes, *Advances in Neural Information Processing Systems*, pp. 5963–5972 (2017).

[7] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K. and Norouzi, M.: Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders, *34th International Conference on Machine Learning*, pp. 1068–1077 (2017).

[8] Bitton, A., Esling, P., Caillon, A. and Fouilleul, M.: Assisted Sound Sample Generation with Musical Conditioning in Adversarial Auto-Encoders, *22nd International Conference on Digital Audio Effects (DAFx-19)* (2019).

[9] Tolstikhin, I., Bousquet, O., Gelly, S. and Schoelkopf, B.: Wasserstein Auto-Encoders, *International Conference on Learning Representations* (2018).

[10] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I. and Welling, M.: Improving Variational Inference with Inverse Autoregressive Flow, *Conference on Neural Information Processing Systems* (2016).

[11] Papamakarios, G., Pavlakou, T. and Murray, I.: Masked Autoregressive Flow for Density Estimation, *31st Conference on Neural Information Processing Systems* (2017).

[12] Kingma, D. P., Rezende, D. J., Mohamed, S. and Welling, M.: Semi-supervised learning with deep generative models, *Advances in Neural Information Processing Systems*, Vol. 4, No. January, pp. 3581–3589 (2014).