Deep Compact Motion Manifold に基づくモーションの生成と編集

木佐 省吾 †　　栗山 繁 †　　向井 智彦 ‡
† 豊橋技術科学大学 情報・知能工学専攻　‡ 首都大学東京 システムデザイン学部
E-mail: †kisa@val.cs.tut.ac.jp, †sk@tut.jp ‡tmki@acm.org

1　はじめに

深層学習を画像や動画の識別だけでなく生成に応用する研究が近年盛んに提案されており，それらの技術はキャラクタ・アニメーションで用いられるモーションの生成と編集にも利用されている．中でも，ニューラルネットワークのオートエンコーダーで学習した潜在変数空間を用いる手法は，動画像の生成や編集だけでなく CG キャラクタ・アニメーションにおけるモーションの対話的な生成や操作に新たな方式をもたらすものとして注目されている．

本研究では，深層学習を用いてモーションの潜在変数空間を構築する新たな枠組みを提案する．その応用例として，キーとなる姿勢の指定により潜在変数を探索してモーションを自動生成する機構や，時間整列等の前処理を必要としない動きの遷移や補間の機構を開発し，特徴を捉えた尤もらしいモーションを生成する性能を検証する．

2　関連研究

Holden らは，オートエンコーダとして構成される，Motion Manifold [1] と呼ばれる潜在変数空間を構築し，その空間での値を歩行の進行方向等の制御変数から選択するもう一つの深層学習を導入して対話的なアニメーションの生成手法を開発した．この手法においては多様体空間の次元数はモーションデータの次元数より拡大しており，その高次元の変数空間での効率的な探索には適していない．また，オートエンコーダー自体も符号器と復号器の各々が単層で構成された極めて単純な構造であるため，生成されるモーションの多様性に関しても疑いがある．本稿で提案する手法は，この Motion Manifold に取って代わる潜在変数空間の新たな構築法を提案するものであり，オートエンコーダを用いる代わりに新たなネットワーク構造を導入する．具体的には，多層のネットワークを用いることによって生じる動きの品質劣化を回避するために敵対的生成ネットワーク [2] の学習を導入し，従来手法よりも低次元な空間に深層学習で多様なモーションを埋め込むことによって，編集の際の計算量や安定性を向上させる手法を提案する．本手法で提案される深層ネットワークで構成される低次元の潜在変数空間を，以後 Deep Compact Motion Manifold（略して **DCMM**）と呼ぶ．

潜在的な変数空間を構築し，モーションの編集に用いる手法は数多く提案されており，その代表例としてはガウス過程に基づく潜在空間を逆運動学計算に適用した手法 [3] や，モーションの動的特性の評価に用いた手法 [4] が挙げられるが，潜在変数の各値は姿勢の状態を表しており，尤もらしいモーションを生成するには，その空間内の自然な軌道を計算する必要がある．一方，本手法における DCMM での値は，固定フレーム長の姿勢系列から成る動きの状態を表すので，モーション単位での合成が可能になる．

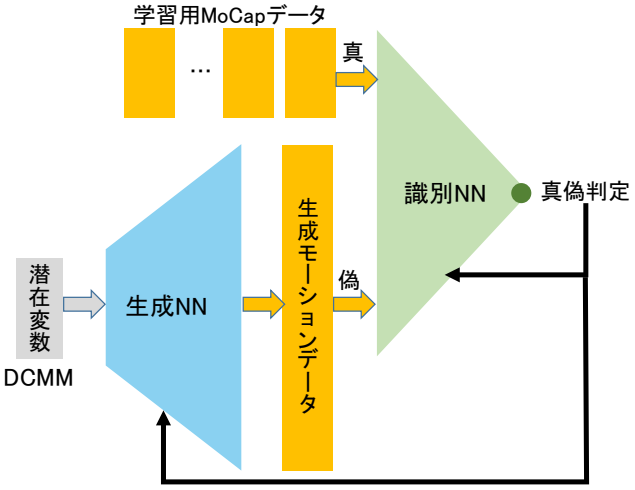

図 1: 提案する敵対的生成ネットワーク (GAN)

3　Deep Compact Motion Manifold

本手法では，敵対的生成ネットワーク（Generative Adversarial Network，以後 GAN）を用いた深層学習を導入するので，図 1 に示す様な生成用と識別用の 2 種類のニューラルネットワークを構成する．本手法においては，生成用のニューラルネットワーク（以後，生成 NN）が DCMM 内からサンプリングされる値からモーションを生成するのに用いられ，識別用のニューラルネットワーク（以後，識別 NN）はネットワークの学習のためだけに用いられる点に留意されたい．Algorithm 1 に示す生成用と識別用のニューラルネットワークの構成と学習法について次節以降に詳述する．

3.1　生成用ニューラルネットワーク（生成 NN）

生成 NN は DCMM 内の変数 $z \in \mathbb{R}^N$ を入力として，図 2 の点線矢印部に相当する変換層を多段階に繋げて構成す

Algorithm 1 学習過程のフローチャート．学習ステップ毎の識別 NN の更新回数（ハイパーパラメータ）は 1 回とする．

for 総学習ステップ数 **do**
  for 識別 NN の更新回数 **do**
    • M 個の潜在変数のミニバッチ $\{\boldsymbol{z}^{(1)},\cdots,\boldsymbol{z}^{(M)}\}$ をガウス分布よりサンプリング
    • M 個の本物データのミニバッチ $\{\boldsymbol{x}^{(1)},\cdots,\boldsymbol{x}^{(M)}\}$ をデータセットよりサンプリング
    • 識別 NN について，式 (2) の損失関数 L_D を最大化するように更新
  end for
  • M 個の潜在変数のミニバッチ $\{\boldsymbol{z}^{(1)},\cdots,\boldsymbol{z}^{(M)}\}$ をガウス分布よりサンプリング
  • 生成 NN について，式 (1) の損失関数 L_G を最小化するように更新
end for

る．最初の段においては，逆畳み込み層を用いて姿勢軸と時間軸の次元数が拡張される．続く段においては時間軸に沿った 1 次元の畳み込み層を用いて，時間軸の次元数を特徴マップサイズとして拡張しながら姿勢軸の次元数を 2 段階で縮退させる．そして，最終的に固定長フレーム数の仮想人間の姿勢ベクトルを出力する．

次元数の拡張には逆畳み込み演算が用いられる事例が多いが，フィルタの刻み幅（stride）の設定方法によっては周期的な雑音成分が発生することが知られている [5]．したがって本手法では，2 段目以降の時間軸での次元数の拡張には逆畳み込み演算ではなく線形のアップサンプリングと畳み込み演算の組み合わせを用いることにした．本ネットワークでは，姿勢軸はチャンネル情報として扱われている点に注意されたい．

最終段を除く各層での出力では LeakyLeRu 関数 [6] を活性化関数に用い，最終段での出力では双曲線正接 (tanh) を用いた．さらに，安定した学習のためには各層の出力で正規化の処理が施されるが，本手法では計算効率が良いとされている Pixelwise Feature Vector 正規化 [7] を用いた．

次章の生成実験においては DCMM の次元数は $N=50$ 次元に設定し，図 2 に示す様な 6 段の層からなる構成を用いた．モーションデータは研究室で構築した通常動作の MoCap のデータセットを用い，意味が十分に捕捉できるサンプリングの解像度と単位となるモーションのフレーム長を考慮して，1 秒間に 10 フレーム（10 fps）のデータを 30 フレーム分（すなわち，3 秒間分）を出力信号の時間軸に沿った次元数とした．一方，後述する様に，モーションの識別 NN を学習させるために用いる MoCap データの関節数は 23 個であり，ルート関節の 6 次元情報を含めた 75 $(=23\times3+6)$ 次元が，出力信号の姿勢軸の次元数となる．

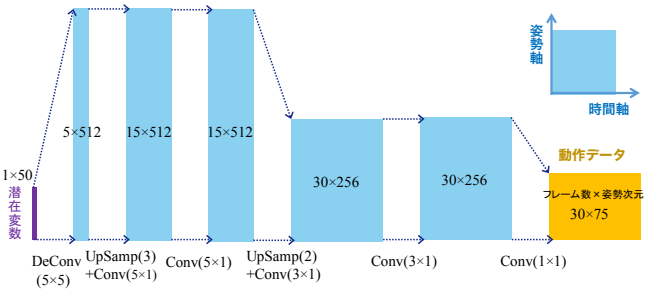

図 2: 生成 NN の構成図．色付きブロック内の数値はデータテンソルの次元数（時間軸×姿勢軸），および [逆] 畳み込み層名（[De]Conv）の後の丸括弧内の数値は，時間軸に沿った（フィルタの幅×刻み幅）を表す．また，アップサンプリング層名（ UpSamp）の後の丸括弧内の数値は拡大スケールの値を示す．

生成 NN の畳み込み／逆畳み込み演算の係数値は，以下の損失関数 L_G を用いて学習される，

$$L_G=\frac{1}{M}\sum_{i=1}^{M}\left[-\log N_D(N_G(\boldsymbol{z}^{(i)}))\right] \tag{1}$$

ただし，M は学習に用いるミニバッチの個数，$\boldsymbol{z}^{(i)}$ は DCMM のガウス分布からサンプリングされる i 番目のデータ，N_G は生成 NN の関数表記，および N_D は N_G を入力とする識別 NN の関数表記である．

3.2 識別用ニューラルネットワーク（識別 NN）

モーションの識別 NN は，図 3 に示す様に多段から構成される畳み込み演算層と，最終段での全結合層で構成される．活性化関数も同じ種類が使用されるが，最終段だけに関してはシグモイドの活性化関数を用いて真偽を判定するスカラー値が出力される．また，本ネットワークも生成 NN と同様に時間軸に沿って特徴量を求めるので，姿勢軸はチャンネル情報として扱われる．

このネットワークの学習には，生成 NN の出力とモーションキャプチャデータ（以後，MoCap データ）から同じフレーム数だけ抜き出した学習用データから計算される 2 種類の信号を交互に入力として用い，MoCap データを真，および生成 NN からの出力を偽と識別する様に，以下の損失関数 L_D を用いて学習させる．

$$L_D=\frac{1}{M}\sum_{i=1}^{M}\left[\log N_D(\boldsymbol{x}^{(i)})+\log\left(1-N_D(N_G(\boldsymbol{z}^{(i)}))\right)\right] \tag{2}$$

ただし，$\boldsymbol{x}^{(i)}$ は MoCap のデータ分布からサンプリングされる i 番目のデータである．

次に，GAN の識別 NN を安定に学習させるために行った事前実験について述べる．具体的には，最近提案された Spectral 正規化 [8] を用いた計算と，Wasserstein GAN-gradient penalty [9]（以後，WGAN-gp）を用いた計算を導入し，性能を実験的に比較した．ここで，各学習ステップ

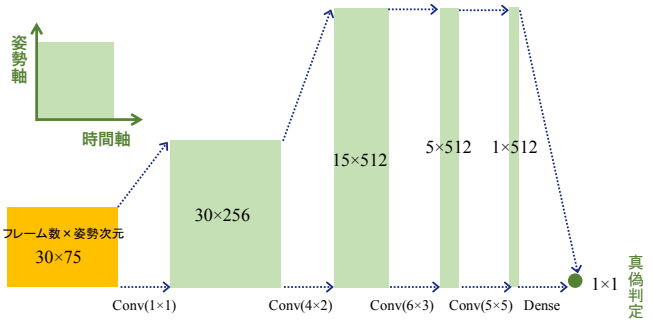

図 3: 識別 NN の構成図．色付きブロック内の数値はデータテンソルの次元数（時間軸×姿勢軸）を，畳み込み層名（ Conv）の後の丸括弧内の数値は時間軸に沿った（フィルタの幅×刻み幅）を，および Dense は全結合層を表す．

での識別 NN の更新回数は，WGAN-gp と Spectral 正規化では各々の推奨値の 5 回と 1 回を設定した．生成 NN と識別 NN の最適化には Adam 法を用い，学習率は経験的に $\alpha = 10^{-4}$ を，二つの勾配係数の値はそれぞれ WGAN-gp では $\beta_1 = 0.0, \beta_2 = 0.9$ Spectral 正規化では $\beta_1 = 0.5, \beta_2 = 0.999$ に設定し，ミニバッチの数は $M = 64$ ，学習回数は 100 万ステップとした．計算機構は $TensorFlow^{TM}$ を用いて実装し，GPU に GTX1080，CPU に Xeon E5-2640 v4 を搭載したワークステーションで計算した結果，WGAN-gp では約 67 時間，Spectral 正規化では約 17 時間の学習を要した．

関節の回転運動の振る舞いに関しては両手法とも品質に差異は認められなかったが，WGAN-gp で生成したモーションには若干の不自然な動きが確認された．したがって，本手法では約 4 倍の計算効率を示し比較的品質の良かった Spectral 正規化を導入する．

4　生成実験

4.1　学習データ

本実験では，前述した様に関節数 23 個の設定で 120 fps で計測された MoCap データを 12 フレームごとに抜き出した 10 fps のデータを 10 フレーム刻みで（ 1 秒間隔）で 30 フレーム分（ 3 秒間）の単位を学習データに用いた．その結果，47 個のファイルから総数 10,511 個の学習データが得られた．ただし，この MoCap データ集合は， 2 名の演者（男性と女性が各 1 名）の，歩行等の移動系を中心とする様々な動きを計測して作成されたものである．

既存手法では関節の位置を入力データとして学習させているが，本手法では MoCap データの標準形式であるオイラー角で表された関節回転角を用いた．関節角を用いる事で四肢の長さの不自然な変動は回避できるが，Gimbal Lock の影響や $\pm\pi$ での値の不連続性での問題が生じる可能性がある．しかしながら，本データセットではこれらの要因による品質の劣化は認められなかった．ただし，オイラー角 θ は $-1 \leq \theta \leq 1$ の範囲に収まる様に線形に正規化を施した値を学習データとし，モーションの生成時には元のスケールに復元する．

ルートの関節に関しては，計測時の条件（初期的な位置と向き）の相違を解消できる普遍的な値を用いる必要がある．本手法では水平面に投影された 2 次元の移動速度ベクトルと垂直軸周りの回転速度，およびルート関節の床面からの高さと身体座標系での水平前頭軸と水平矢状軸に関する向きを表す回転量（オイラー角）の 6 次元で構成する．ゆえに，ルート関節の絶対的な位置や向きは，速度に関する値を積分計算によって求める．これらの値に関しても，学習の安定性確保のために，全ての値 Θ が $-1 \leq \Theta \leq 1$ の範囲に収まる様に線形に正規化を施すので，モーションの生成時には元のスケールに戻す必要がある．既存手法 [1] ではルート関節の向きは左右の肩と股関節の位置から推定しているが，本手法は計測時の値をそのまま学習させているので，より真正な値を生成できる可能性がある．

4.2　キー姿勢を用いた動作の生成

既存のキーフレーム・アニメーションでは，動きに含まれる典型的な姿勢をキー姿勢として与え，それらをスプライン関数などの滑らかな曲線で補間してアニメーションを生成している．本手法においては，DCMM 内の 1 点から単位フレーム長のモーションを生成できるので，キー姿勢が含まれる DCMM 空間の座標を探索することにより，その姿勢を含みながら学習データの特徴を反映した尤もらしい動きを生成することができる．

DCMM 空間の潜在変数 $\boldsymbol{z}$ に対して，与えられるキー姿勢の姿勢ベクトル $\boldsymbol{P}_i,\ i = 1, 2, \ldots, k$ とすると，

$$\boldsymbol{z}^* = \arg\min_{z} \sum_{k} \min_{f} \left\| N_G^f(\boldsymbol{z}) - \boldsymbol{P}_k \right\|$$

を満たす潜在変数 $\boldsymbol{z}^*$ を求める．ただし，$\| \ \|$ はユークリッドノルムであり，$N_G^f(\boldsymbol{z})$ は，$\boldsymbol{z}$ より計算されるモーション（姿勢系列）の f 番目の姿勢ベクトルを表す．

多層のニューラルネットワークで構成される関数 N_G は高度に非線形なので，局所解に陥りにくい探索方法が適している．本手法では，微分計算を必要としないヒューリスティックな解探索を行う Covariance Matrix Adaptation Evolution Strategy [10] を用いた．この手法は標本値を並列的に探索して解空間を絞り込むので，ニューラルネットワークの並列計算を効率的に活用できる．

今回の実験では，一度の更新計算で発生させる個体数は 10 個とし，探索時の標準偏差は 1.0 に設定した．また，最大の繰り返し回数は 1000 回としたが，多くの場合はそれ以下の繰り返し数で誤差が一定値 (0.01 程度) 以下に収ま

り計算を終了させた．計算は時間は 1000 回の繰り返しでも 10 秒程度であった．

一方，Motion Manifold [1] はオートエンコーダを介して構成されるので，そのエンコーダ部の入力に同じフレーム長の MoCap データを入力すれば対応する潜在変数の値が取得できる．しかしながら，本手法のように少数の姿勢データをエンコーダの入力とすることは不可能であり，高次元の空間内では適切な潜在変数を探索するのは困難であることが予想される．実際に，今回の実験に用いたデータに合わせた 3840 次元の Motion Manifold を同じ条件で学習させて探索計算を実行した結果，約 40 分を要した 10000 回の繰り返し計算後でも収束には至らずに異常な動きが生成された (いずれも，iMac Pro 3.2 GHz Intel Xeon W での計算結果)．

添付する動画に，様々なキー姿勢を指定して生成されるモーションの例を示す．ただし，キー姿勢は学習に用いた MoCap データから抜き出して使用し，その MoCap データの動きと類似の動きが生成されるかを検証した．キー姿勢が特徴的であれば，一つの姿勢をしているだけでも類似した動作が生成されているが，数値的な探索計算は毎回異なる結果を算出するので，場合によっては似ていないモーションが生成される場合もある．しかしながら，複数の姿勢を指定すれば，所望するモーションが精度よく生成されることが確認された．この複数のキー姿勢は連続するフレームでの値のみではなく，任意の間隔で指定した場合にでも有効に働くことが確認される．多くの試行実験の結果，キー姿勢は 3 個が指定されれば十分な精度が得られる事が確認できた．

学習の汎化性能を検証するために，学習データとして用いなかった MoCap データから抜粋したキー姿勢による生成を試みた．学習時には含まれないデータなので，元の動きを再現することは本質的に困難であり動きには不自然さが認められるものの，全体的な挙動としての類似性は見られた．

4.3 モーション間の補間と遷移

前述したキー姿勢で生成したモーションを連結させたり，進行方向や歩容の異なる多くの類似動作を繁殖させたりするために，モーションデータの空間軸および時間軸に沿った補間方法が一般的に用いられる．本手法においては，このモーションの補間も DCMM の潜在変数空間で計算が可能になる．

キー姿勢によって生成された二つのモーション $\mathbf{m}_0$ と $\mathbf{m}_1$ の潜在変数の値を各々 $\mathbf{z}_0, \mathbf{z}_1$ とすると，その補間値を用いて生成されるモーションは両モーションの中間的な特徴を有することが期待される．中間的な潜在変数 $\tilde{\mathbf{z}}(t)$ の算出には，ガウス分布として張られる多様体空間内での補間計算に適していると報告 [11] されている，以下の式で与えられる球面線形補間を用いる．

$$\begin{aligned} \tilde{\mathbf{z}}(t) &= \frac{\sin(1-\bar{t})\vartheta}{\sin\vartheta}\mathbf{z}_0 + \frac{\sin\bar{t}\vartheta}{\sin\vartheta}\mathbf{z}_1 \,, \\ \vartheta &= \arccos\left(\frac{\mathbf{z}_0\cdot\mathbf{z}_1}{|\mathbf{z}_0||\mathbf{z}_1|}\right),\ \bar{t} = \frac{t-1}{T-1} \in [0,1] \,, \end{aligned}$$

ただし，t は時間軸でのフレームの値を表す．

二つのモーションの中間的なモーションを生成（すなわち，補間）する場合には，一つの中間的な変数値 $0 < t_c < T$ に対するモーション $\tilde{\mathbf{m}}(t_c)$ を生成すれば良い．一方，二つのモーションを遷移させる場合には，区間 T での全モーション $\tilde{\mathbf{m}}(t), \forall t = 0, 1, 2, \ldots, T$ を生成し，$\tilde{\mathbf{m}}(t)$ に対して t 番目のフレームの姿勢を抜き出して繋ぎ合わせた姿勢系列が遷移モーションとして構成される．

添付動画に示される中間動作と遷移動作により，尤もらしい動きが生成されていることが確認できる．特に，通常では補間が困難と考えられる，前向きから後向きと横向きから前向きの歩行への遷移や，左右の脚を出す順番が逆になった歩行同士の補間に対しても，モーションデータ間の整列を必要とせずに自然な動きが柔軟に生成されている点は，本提案手法の優れた性能を示している．

5 おわりに

本稿では，敵対的生成ネットワーク（GAN）で学習される DCMM を用いたモーションデータの生成と，その値を介したモーションの簡易な生成・編集方法を提案した．既存手法よりも深層のネットワークで安定かつ高品位にモーションを生成する機構を構築し，その表現力の拡大を達成できた．また，潜在変数の次元数も既存手法の 1.3% 程度にまで削減できたので，潜在変数空間での数値演算に基づくモーションの効率的な探索や，補間への応用を実現できた．また，既存手法では関節の位置を学習対象として開発が進められてきたが，関節角度を直接学習対象としても高い精度が得られる事を示した．

本手法による，モーションに対する潜在変数空間を用いた操作・編集の方法は，他の潜在空間を用いた既存の方法も導入することができると考えられる．今回に示した機能以外にも応用範囲を広げるためには，足跡や進行方向などの制御変数を用いた対話的なモーションの生成機構を，新たなネットワークを学習させて統合させる必要がある．また，DCMM の変数を用いてモーションのスタイル（動きの個人的な特徴や癖など）の転写を組み込む事も考えられる．さらに，人間な自然の動作以外にも，ゲーム用に編集された誇張のある動きに対する本手法の有効性に関しても，今後調査を進めていく．

DCMM の 50 次元という数は経験的に設定したが，与えられたモーションデータセットに含まれる全特徴の表現力

を失うことなく，出来るだけ少ない次元数を見積もる方法に関しても，今後の調査が必要である．GAN で生成されるデータの多様性を数値的に評価する手法 [12] も提案されているので，多様性という観点から最適な次元数を推定できる可能性がある．

今回の実験では Motion Manifold を比較対象としたが，既存のオートエンコーダに基づく手法との比較も必要である．今後は，単純な実装方法や潜在変数が正規分布となるような識別器を導入した実装方法等と比較して，GAN に基づく手法の優位性を検証する予定である．

本手法によるニューラルネットワークに基づいたモーション生成は，既存手法と同様の欠点や限界も有する．改良が最も望まれるのは，十分ではないルート関節の状態の推定精度を高めることである．ルート関節の位置と向きは，本質的には下半身の姿勢に加えて足先および踵の接地状態を推定して逆運動学を用いた計算と一致する必要がある，この値が正確に推定できなければ足先の地面に対する不自然な状態（陥没や滑り等）が目立ってしまう．これらの誤差は擬似逆行列等を用いてフレーム毎に数値的に補正することも可能であるが，接地状態が不安定な歩行や足先の遊離期間の長い動きに対しては適用が困難である．これらの問題を解決するために，ルート関節の状態の推定機構を個別に構築する様なアプローチを現在検討中である．モーションデータは画像データと比較して次元数が小さいので，深層学習に要する計算時間は膨大なものとはならないが，階層的な学習法などの，より効率的かつ安定な学習法も再検討する予定である．

参考文献

[1] D. Holden, J. Saito, and T. Komura, “A deep learning framework for character motion synthesis and editing,” *ACM Trans. Graph.*, vol. 35, no. 4, pp. 138:1–138:11, Jul. 2016.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[3] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović, “Style-based inverse kinematics,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 522–531, Aug. 2004.

[4] J. M. Wang, D. J. Fleet, and A. Hertzmann, “Gaussian process dynamical models for human motion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 283–298, Feb 2008.

[5] A. Odena, V. Dumoulin, and C. Olah, “Deconvolution and checkerboard artifacts,” *Distill*, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[6] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” 2013.

[7] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” in *International Conference on Learning Representations*, 2018.

[8] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *International Conference on Learning Representations*, 2018.

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” *CoRR*, vol. abs/1704.00028, 2017. [Online]. Available: http://arxiv.org/abs/1704.00028

[10] N. Hansen and A. Ostermeier, “Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation,” in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 312–317.

[11] T. White, “Sampling generative networks: Notes on a few effective techniques,” *CoRR*, vol. abs/1609.04468, 2016.

[12] S. Arora, A. Risteski, and Y. Zhang, “Do GANs learn the distribution? some theory and empirics,” in *International Conference on Learning Representations*, 2018.