社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

多話者音声合成のための Adversarial Regularizer を考慮した学習アルゴリズム

仲井佑友輔† 　齋藤　佑樹† 　宇田川健太† 　猿渡　　洋†

† 東京大学　〒113–8656 東京都文京区本郷 7-3-1

あらまし　本稿では，Adversarial Reguralizer を考慮した敵対学習による多話者音声合成モデルを提案する．従来法では，識別的なタスクによって事前学習した Speaker Encoder から目的話者の話者埋め込みを抽出し，音声合成ネットワークに入力を行う．しかし，学習された話者埋め込みの分布する特徴量空間は音声合成ネットワークにとって必ずしも解釈性が高いとは限らず，未知話者の話者埋め込みを上手く抽出できる保証が無いという問題があった．提案法では，事前学習済みの話者埋め込み空間をうまく解釈できる音声合成ネットワークの構築を目的とし，学習アルゴリズムとして Adversarial Reguralizer を考慮した敵対学習を提案する．提案法では，話者埋め込みを混合して合成した音声の特徴量と，自然音声の特徴量が識別不可能となるような正則化項を考慮して音声合成ネットワークを学習する．実験的評価により，提案法が合成音声の話者類似性と，話者モーフィングの操作性を改善する傾向にあることを示す．

キーワード　DNN 音声合成，転移学習，敵対学習，話者モーフィング，話者埋め込み

Yusuke NAKAI†, Yuki SAITO†, Kenta UDAGAWA†, and Hiroshi SARUWATARI†

† The University of Tokyo, Hongo 7–3–1, Bunkyo-ku, Tokyo, 113–8656 Japan

1. はじめに

音声研究の一分野であるテキスト音声合成 (text-to-speech: TTS) [1] は，入力されたテキストに話者の特徴などの非言語情報を与え，自然な音声を合成する技術であり，医療シーン [2], [3] から言語教育 [4] に至るまで多様な応用が提案されている．近年はディープニューラルネットワーク (deep neural network: DNN) による学習手法が発展しており，日夜新しい手法が提案されている [5]．TTS において，複数話者の音声合成を目的とした技術は多話者テキスト音声合成 [6] と呼ばれる．

DNN による多話者テキスト音声合成では，音声データから固定長のベクトルで表される話者の特徴量（話者埋め込み）[7], [8] を抽出し，これを音声合成ネットワークに条件付けすることで音声合成を行う．Jia ら [9] は音声合成ネットワークから独立した Speaker Encoder によって，話者埋め込みを事前学習させる多話者音声合成の手法を提案した．この手法において，Speaker Encoder は話者識別的なタスクにより学習を行う．埋め込み空間上で，同一話者の発話に対応する話者埋め込みは近くに，異なる話者の発話による話者埋め込みは遠くに分布するように学習を行い，十分な学習がなされた Speaker Encoder は話者の特徴を上手く抽出できることが期待される．Speaker Encoder は音声合成ネットワークとは独立に学習されるため，ネットワーク学習時に話者埋め込みを更新する必要が無く，少量の音声データでも高品質な音声が合成できるという利点がある．しかし，Speaker Encoder によって学習された話者埋め込み空間は音声合成ネットワークとは独立に学習されており，必ずしも音声合成ネットワークにとって解釈性が高い空間であるとは限らない．話者埋め込み空間の解釈性の低さは，学習データに含まれない未知話者の合成音声の品質劣化を生じさせるのみならず，合成音声の話者性を制御する際の障壁となり得る．

本稿では，事前学習済みの話者埋め込み空間をうまく解釈できる音声合成ネットワークの実現を目的として，Adversarial Regularizer を考慮した敵対学習アルゴリズムを提案する．提案法ではまず，異なる話者による発話から抽出した話者埋め込みをランダムな割合で混合し，混合された話者埋め込みからメルスペクトログラムを合成する．その後，合成されたメルスペクトログラムは Critic という DNN に入力され，Critic はメルスペクトログラムから混合率の推定を行う．一方で音声合成ネットワークは，Critic が混合された話者埋め込みによるメルスペクトログラムに対しても，真の話者埋め込みによるものであると推定するように学習を行う．これにより，混合された話者埋め込みから合成される未知話者の音声特徴量と，自然音声特徴量の区別がつかなくなるような正則化が実現されるため，話者埋め込み空間の解釈性向上が期待できる．実験的評価により，提案法が合成音声の話者類似性と，話者モーフィングの操作性を改善する傾向にあることを示す．

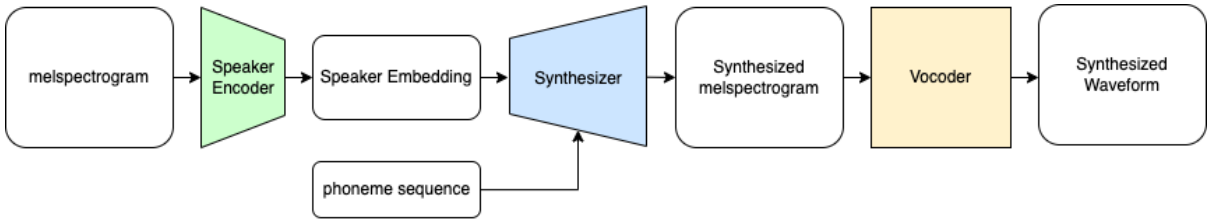

図 1　Speaker Encoder を転移学習させる多話者音声合成モデルの概略図．Speaker Encoder と Vocoder のモデルパラメータは，Synthesizer の学習時には更新されない．

2. 従来の多話者音声合成

本節では，Jia らの先行研究 [9] で提案された多話者音声合成の手法を説明する．

2.1 多話者音声合成ネットワーク

多話者音声合成ネットワークは，Synthesizer と Vocoder の２つの DNN から構成される．Synthesizer は話者埋め込みとテキストを入力として，メルスペクトログラムを出力する．Vocoder はメルスペクトログラムから音声を生成する．モデル学習時，これらは独立に学習される．

2.2 話者識別モデルの転移学習

2.2 節では，本稿のベースラインとなる，Speaker Encoder を事前学習する多話者音声合成モデルについて概説する．このモデルにおいて，Synthesizer に入力される話者埋め込みは Speaker Encoder という DNN によって事前学習される．図 1 にモデルの概略図を示す．

Speaker Encoder は音声波形から対応する話者埋め込みを抽出し，Generalized end-to-end (GE2E) 損失 [8] を最小化するように学習する．GE2E 損失を最小化する学習は識別的なタスクであり，話者埋め込み空間において，同じ話者の埋め込みは近くに，異なる話者の埋め込みは遠ざかるように学習される．

学習された話者埋め込みは，合成音声の話者性を制御するために Synthesizer に入力される．Synthesizer は，話者埋め込みに対応する話者の音声の特徴量（メルスペクトログラムなど）の予測誤差を最小化するように学習する．この際，Speaker Encoder のモデルパラメータは事前学習済みのもので固定し，Synthesizer の学習時には更新しない．以降，この Synthesizer の学習時の損失関数を $\mathcal{L}_{\text{TTS}}$ と表記する．推論時には，まず，合成したい話者の参照音声を Speaker Encoder に入力し，話者埋め込みを抽出する．次に，この話者埋め込みと合成したいテキストを Synthesizer に入力し，合成音声の特徴量を予測する．最後に，この特徴量を Vocoder に入力して合成音声の波形を生成する．

2.3 従来法の問題点

従来法では，Synthesizer に入力される話者埋め込みは識別的タスクのみによって学習されたものであるため，話者埋め込み空間が Synthesizer にとって意味のある空間になりうる保証が無い．そのため，未知話者について音声合成を行う際，抽出した話者埋め込みが Synthesizer にとって解釈性の高いベクトルになるとは限らず，目的話者の話者性が十分に再現できないことが考えられる．

3. 提案する多話者音声合成

従来の多話者音声合成モデルの問題点に対し，本稿では Adversarial Regularizer を考慮した敵対学習によるモデルを提案する．

提案法では，音声合成ネットワークの他に Critic というネットワークを追加する．Critic は二名の話者埋め込みの混合から合成されたメルスペクトログラムを入力とし，その混合率を出力するような学習を行う．一方で音声合成ネットワークは，Critic が入力されたメルスペクトログラムを真の話者埋め込みによるものだと推定させるように学習する．

3.1 Adversarial Regularizer

データの特徴量抽出に用いられる DNN として AutoEncoder [10] というモデルが知られている．AutoEncoder は特徴量抽出・データ再構成によりデータの持つ特徴を低次元の特徴量として得るモデルだが，それらの特徴量が分布する空間が解釈性の高いものかどうかは定かではない．これに対し，抽出した特徴量同士を補間することでデータの自然なモーフィングを実現し，解釈性の高い特徴量空間を得るために考案されたのが Adversarial Regularizer を考慮したモデルである．このモデルでは，AutoEncoder の他に Critic というネットワークを追加し，敵対学習を行う．まず，Encoder は２つの入力データ $\boldsymbol{x}_1, \boldsymbol{x}_2$ から特徴量を抽出する．次に，得られた特徴量 $\boldsymbol{z}_1 = f_\theta(\boldsymbol{x}_1), \boldsymbol{z}_2 = f_\theta(\boldsymbol{x}_2)$ をランダムな割合 $\alpha \in [0, 0.5]$ で線形に混合する．最後に，混合された特徴量 $\alpha\boldsymbol{z}_1 + (1-\alpha)\boldsymbol{z}_2$ を Decoder に入力し，データ $\hat{\boldsymbol{x}}_\alpha = g_\phi(\alpha\boldsymbol{z}_1 + (1-\alpha)\boldsymbol{z}_2)$ を得る．一方 Critic d_ω は，Decoder からの入力 $\hat{\boldsymbol{x}}_\alpha$ に対して，データが混合されたものであった場合はその混合率 α を，混合されていないデータであった場合は $\alpha = 0$ を出力するよう学習する．

AutoEncoder と Critic の損失関数 $\mathcal{L}_{f,g}, \mathcal{L}_d$ はそれぞれ以下のように定義される．

$$\mathcal{L}_{f,g} = \|\boldsymbol{x}_1 - g_\phi(f_\theta(\boldsymbol{x}_1))\|^2 + \lambda\|d_\omega(\hat{\boldsymbol{x}}_\alpha)\|^2 \tag{1}$$

$$\mathcal{L}_d = \|d_\omega(\hat{\boldsymbol{x}}_\alpha) - \alpha\|^2 + \|d_\omega(\gamma\boldsymbol{x}_1 + (1-\gamma)g_\phi(f_\theta(\boldsymbol{x}_1)))\|^2 \tag{2}$$

AutoEncoder の損失関数 $\mathcal{L}_{f,g}$ の第一項は AutoEncoder の入力に用いたデータと再構成されたデータとの二乗誤差である．第二項は，混合したデータに対しても真のデータから再構成したものであると Critic に判別させるための損失である．つまり

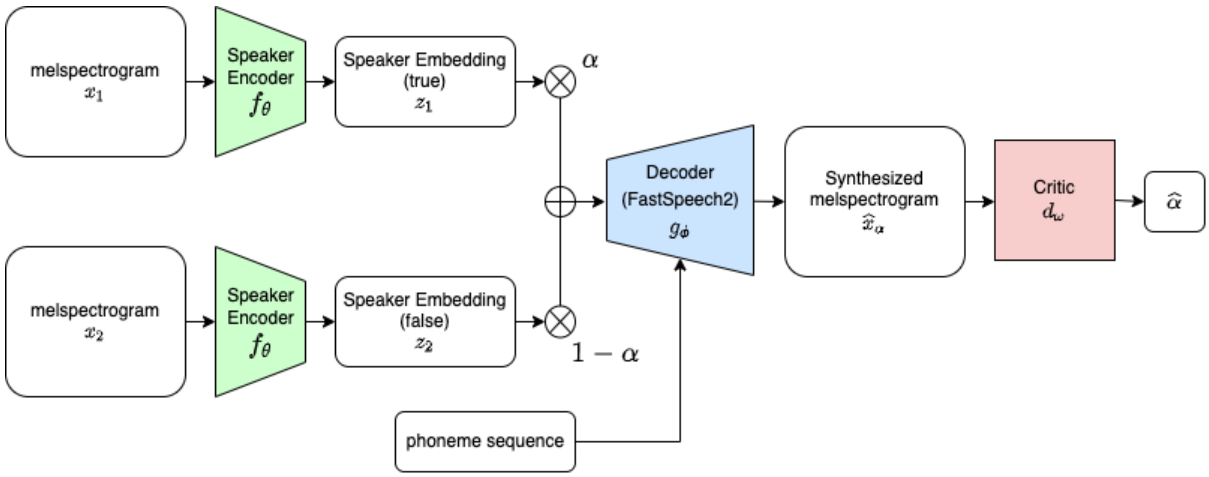

図 2　提案する敵対学習における混合話者の音声特徴量生成と，Critic による混合率推定の概略図.

$d_\omega(\hat{\boldsymbol{x}}_\alpha) = 0$ と出力されるように Critic を欺くための正則化項である．λ は第二項の重みを調整するハイパーパラメータである．

Critic の損失関数 $\mathcal{L}_d$ の第一項は Critic が予測した混合率 $d_\omega(\hat{\boldsymbol{x}}_\alpha)$ と真の混合率 α との二乗誤差である．第二項は AutoEncoder の再構成データの精度が低い学習の初期段階を安定させるための項である．この項は入力のデータと再構成されたデータをハイパーパラメータ $\gamma \in (0, 1)$ の割合で混合したデータを Critic に入力した際の α の予測値の二乗で表される．

3.2 Adversarial Regularizer を考慮した多話者音声合成の学習法

提案法における，Synthesizer と Critic との敵対学習の概略図を図 2 に示す.

提案モデルでは，まず各バッチから異なる話者のメルスペクトログラムの組 $\boldsymbol{x}_1, \boldsymbol{x}_2$ を用意し，それぞれ学習済みの Speaker Encoder f_θ に入力する．出力された話者埋め込み $\boldsymbol{z}_i = f_\theta(\boldsymbol{x}_i)$ $(i \in \{1, 2\})$ について，$\boldsymbol{z}_1$ を真の話者埋め込み，$\boldsymbol{z}_2$ を混合に用いる偽の話者埋め込みとおく．バッチごとに異なる話者の組と話者埋め込みを選択するアルゴリズムを **Algorithm 1** に示す．次に，これらをランダムな割合 $\alpha \in [0, 0.5]$ で混合し，混合された話者埋め込み $\boldsymbol{z}_{\text{mixed}}$ を Synthesizer g_ϕ に入力する．最後に，出力された混合話者のメルスペクトログラム $\hat{\boldsymbol{x}}_\alpha = g_\phi(\boldsymbol{z}_{\text{mixed}})$ を Critic d_ω に入力する．Critic は入力されたメルスペクトログラムが混合話者のものであった場合はその混合率 α を，そうではない場合は $\alpha = 0$ を出力するよう学習する．Critic の損失関数は，式 (2) と等価である．モデルの更新順序として，まず Critic を更新し，次に Synthesizer に関して式 (3) で与えられる損失関数を定義し，Synthesizer の更新を行う.

$$\mathcal{L} = \mathcal{L}_{\text{TTS}} + \lambda \| d_\omega(\hat{\boldsymbol{x}}_\alpha) \|^2 \tag{3}$$

これは，従来の音声合成の予測誤差と，前節で述べた adversarial regularizer の損失のマルチタスク学習と解釈できる.

Algorithm 1 混合に用いる真の話者埋め込み，偽の話者埋め込みをバッチ単位で抽出するアルゴリズム

```
batch から speakerlistを抽出
fake speakerlist = []
for speaker ∈ speakerlist do
    different speakers = []
    for speaker_2 ∈ speakerlist do
        if speaker = speaker_2 then
            continue
        end if
        speaker_2を different speakers に追加
    end for
    different speakerを different speakers からランダムに選択
    fake speakerlistに different speakerを追加
end for
speaker listに対応する true embeddings を抽出
fake speaker listに対応する fake embeddings を抽出
```

4. 実験的評価

4.1 実 験 条 件

宇田川らの先行研究 [11] を参考に，Speaker Encoder の学習には日本語話し言葉コーパス (Corpus of Spontaneous Japanese: CSJ) [12] を用いた．CSJ コーパスは日本人 1417 名（男性話者 947 名，女性話者 470 名）の計 660 時間の発話データを含む．CSJ の音声データは 16kHz にダウンサンプリングし，フレームシフトは 10ms とした.

Synthesizer の学習には Japanese versatile speech コーパス (JVS) [13] のパラレル発話データである voiceactress100 を使用した．voiceactress100 は日本人話者 100 名（男性話者 49 名，女性話者 51 名）の計 22 時間の発話データ（話者ごとに 100 発話）を含む．学習には話者 96 名分の発話データを使い，学習データに含まれていない 4 話者 (“jvs005", “jvs010", “jvs060", “jvs078") は [11] に準拠して選定し，これらを評価用話者とした．評価用話者の発話を含まない計 9600 発話のデータをランダムにシャッフルし，うち 512 発話を検証用，残りを学習用データとして使用した．voiceactress100 には発話内容と発話ラベルが一致しないデータや，収録の失敗により音声が極端に短

いデータが含まれるため，それらを前処理として取り除いた．評価話者のうち “jvs060" の発話数は前処理により 99 であり，それ以外の評価話者の発話数は 100 であった．JVS の音声データは 22.05kHz にダウンサンプリングし，フレームシフトは 12ms とした．

Speaker Encoder のモデルにおいて， Long short-term memory (LSTM) [14] の最終層の隠れ状態に続く 256 次元への全結合層の活性化関数には ReLU [15] を用い，その後 L_1 正規化を用いて 256 次元の話者埋め込みを推論した．話者埋め込みの学習に用いる GE2E 損失にはソフトマックス損失を用いた．音声合成ネットワークには Wataru-Nakata により公開されている FastSpeech2 [5] のオープンソース実装[1]を用いた．Critic には畳み込みニューラルネットワーク (Convolution Neural Network: CNN) を用いた．畳み込み層は 5 層とし，層間の活性化関数には Leaky ReLU [16] を用いた．畳み込み層から出力されたデータを時系列方向に圧縮，シグモイド関数による正規化を行なったものを出力とした．

FastSpeech2 と Critic は同時に学習を行なった．学習時の最適化には Warmup [16] を用いて学習率スケジューリングを行なった Adam [17] を用い，Warmup step は 4000，学習率の初期値は 0.0625 とした．バッチサイズは 8，学習ステップは 20000 とした．提案法において，敵対学習で用いられるハイパーパラメータ γ, λ には $(\gamma, \lambda) = (0.1, 0.01)$ を用いた．従来法の学習時では $(\gamma, \lambda) = (0, 0)$ とし，損失関数における Adversarial Regularizer を考慮する項を除外した．従来法と提案法それぞれでモデルの学習，評価を行なった．

4.2 客観評価

評価話者 4 名に対して，自然音声の分散情報と音声合成に用いた分散情報との平均二乗偏差 (Root Mean Squared Error: RMSE) を従来法，提案法について計算し，比較を行なった．音声の分散情報として，FastSpeech2 で学習に利用されている特徴量であるピッチ (pitch)，エネルギー (energy), 音素継続長 (duration) を使用した．各分散情報は音素単位で抽出したものである．一般に，正解データ列 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N$ とその予測 $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_N$ との RMSE は以下の式で計算できる．

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2} \tag{4}$$

pitch, energy に関しては, 音素の継続時間を考慮し, duration を二乗誤差の重みとした場合の RMSE の計算も行なった．duration $\boldsymbol{d}_i$ による重み付き RMSE は以下のように計算した．

$$\mathrm{RMSE}_{\mathrm{weighted}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i) \odot (\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i) \cdot \boldsymbol{d}_i} \tag{5}$$

表 3 にそれぞれの分散情報の RMSE の結果を示す．全ての分散情報について従来法のモデルより提案法の RMSE が小さくなっており，合成音声の分散情報に関しては提案法が従来法よりも良好な予測をしていることがわかる．

表 1　自然音声の pitch, energy, duration とモデルから推測した pitch, energy, duration との RMSE．“weighted" と示されているものは正解音声の duration を重み付けして計算したことを表す．

手法	pitch	pitch (weighted)	energy	energy (weighted)	duration
従来法	3.10	9.06	3.64	9.05	25.9
提案法	2.71	8.12	3.53	8.79	25.2

表 2　未知話者での合成音声についての自然性の比較結果．太字は手法間に $p < 0.05$ の有意差があったことを示す．

話者	従来法	提案法
“jvs005"	0.528	0.472
“jvs010"	**0.640**	**0.360**
“jvs060"	0.460	0.540
“jvs078"	**0.568**	**0.432**

表 3　評価話者の合成音声についての話者類似性の比較結果．太字は手法間に $p < 0.05$ の有意差があったことを示す．

話者	従来法	提案法
“jvs005"	**0.560**	**0.440**
“jvs010"	**0.372**	**0.628**
“jvs060"	**0.416**	**0.584**
“jvs078"	**0.392**	**0.608**

4.3 主観評価

Lancers[2] でのクラウドソーシングにより，提案法と従来法のモデルで合成した評価話者 4 名の発話について，三種類の主観評価によって手法間の差異を調べた．

4.3.1 未知話者での合成音声の品質の主観評価

従来法と提案法それぞれのモデルで合成した音声の自然性と話者類似性を，どちらの音声がより自然か，もしくは当該話者に似ているか比較実験を行なった．自然性の評価ではプリファレンス AB テストを，話者類似性の評価では参照音声 (X) に当該話者の自然音声を用いたプリファレンス XAB テストを実施した．自然性の評価と話者類似性の評価それぞれにおいて，評価者は 25 名であり，各評価者はランダムに 10 個の音声サンプル対を評価した．合計の評価セット数は $25 \times 10 = 250$ ずつであった．表 2 に各手法の自然性の評価結果，表 3 に各手法の話者類似性の評価結果を示す．

表 2 について，“jvs005", “jvs060" の合成音声は従来法と提案法で有意差は見られず，“jvs010", “jvs078" については従来法の自然性が提案法を上回る結果となった．これは提案法において，Synthesizer の損失関数 (4.1) には Critic のための損失項が含まれているため，自然音声との二乗誤差のみで学習を行なった従来法での合成音声の方が高い自然性を与えたと考えられる．また，Critic が判別するのは混合された 2 話者の話者埋め込みの混合率のみであり，学習に音声の自然性に関する項が含まれていないため，自然性の学習が困難であったことも原因として考えられる．一方，表 3 を見ると，提案法では “jvs010", “jvs078" における合成音声の話者類似性が有意に向上したことが分かっ

（注1）：https://github.com/Wataru-Nakata/FastSpeech2-JSUT

（注2）：https://www.lancers.jp/

表 4　評価話者の話者埋め込みを混合した合成音声についての自然性の主観評価結果．太字は手法間に $p < 0.05$ の有意差があったことを示す．

混合した話者の組	従来法	提案法
“jvs005", “jvs010"	0.508	0.492
“jvs005", “jvs078"	**0.428**	**0.572**
“jvs010", “jvs060"	0.516	0.484
“jvs078", “jvs060"	**0.404**	**0.596**

た．これは，Adversarial Regularizer を考慮した学習を行い，解釈性の高い特徴量空間を得たことで，未知話者の特徴量抽出の精度が従来法より向上したためと推察される．

4.3.2　補間した話者埋め込みによる合成音声の自然性主観評価

話者埋め込み空間の頑健性の高さを主観評価により確認する．評価用話者のうち，2 名の話者埋め込みを等しい割合で線形に混合し，従来法と提案法それぞれのモデルで音声合成を行なった．各手法の合成音声について，どちらの音声がより自然か，比較実験を行なった．評価用話者の組は男性同士の組 (“jvs005", “jvs078")，女性同士の組 (“jvs010", “jvs060") ，異性同士の組として学習データの話者との主観的類似度 [18] の高い組 (“jvs060", “jvs078")，低い組 (“jvs005", “jvs010") の計 4 組とした．評価者は 25 名であり，各評価者はランダムに 10 個の音声サンプル対を評価した．合計の評価セット数は $25 \times 10 = 250$ であった．表 4 に各手法のスコアの結果を示す．表 4 より，一部の話者の組・手法において提案法が従来法を上回る自然性となった．提案法は先述の通り，自然性において従来法に劣る傾向があるが，一方で特徴量空間の解釈性は向上していると考えられる．そのため，有意差が出た音声に関しては，提案法の特徴量空間の解釈性の高さが自然性の低さを上回ったと考察できる．

4.3.3　話者モーフィングの操作性の主観評価

2 名の評価用話者の話者埋め込みを $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ の割合で線形に混合し，これらから合成された音声による話者モーフィングの操作性についての評価を行なった[3]．受聴者はまず 2 名の評価話者（話者 A, 話者 B とおく）について，それぞれの話者埋め込みによる合成音声を聴く．次に，話者 A, 話者 B の話者埋め込みを α の割合で混合した話者埋め込みによる合成音声を聴き，スライドバーを使って 5 段階（1 : 間違いなく話者 A の音声である（$\alpha = 0$ に対応）〜 5 : 間違いなく話者 B の音声である（$\alpha = 1$ に対応））で評価し，これを一つのタスクとする．評価用話者の組は前小節と同じ 4 組で行なった．評価者は 50 名であり，各評価者はランダムに 20 個のタスクを行なった．合計の評価セット数は $50 \times 20 = 1000$ であった．最後にそれぞれの評価結果についてスコアの平均をとり，正規化を行なった．正規化したスコア $\hat{\alpha}$ は α の推測結果とみなせる．この推測結果 $\hat{\alpha}$ と真の混合率 α との RMSE を計算した．表 5 に各手法の RMSE の結果を示す．表 5 より，全ての話者の組において従来法より提案法がより自然な話者モーフィングを実現していることが分かり，Adversarial Regularizer を考慮した学習を行なったことで，未知話者の話者埋め込みの補間に対しても有効な埋め込み空間を学習できたことを示唆していると考えられる．

(注3)：主観評価に用いた音声のサンプルは https://bit.ly/3AKO6AW から確認できる．

表 5　混合率を変化させた音声による話者モーフィングの操作性の評価値と真の混合率との　RMSE．太字は従来法より提案法が RMSE が小さくなった（より真の混合率に近い評価結果になった）ことを表す．

混合した話者の組	従来法	提案法
“jvs005", “jvs010"	0.110	**0.0858**
“jvs005", “jvs078"	0.0985	**0.0646**
“jvs010", “jvs060"	0.0710	**0.0701**
“jvs078", “jvs060"	0.118	**0.0941**

5. おわりに

本稿では，Adversarial Regularizer を考慮した敵対学習による多話者音声合成モデルを提案し，実験的評価により手法の有効性を検証した．実験の結果，合成音声の主観的な自然性に関して改善の余地があるが，話者類似性では従来法を上回る精度の音声を合成できることが分かった．また，従来法より話者モーフィングの操作性が改善されたことも確認された．今後は，本手法で新たに導入している Critic の構造や話者埋め込みの補間方法の吟味を行い，より解釈性の高い特徴量空間の構築を検討する．

謝辞: 本研究は，JST ムーンショット型研究開発事業，JPMJMS2011（アルゴリズム開発），JSPS 科研費 21K21305（実証実験）の支援を受けたものです．

文　献

[1] Y. Sagisaka: “Speech synthesis by rule using an optimal selection of non-uniform synthesis units”, Proc. ICASSP, New York, U.S.A., pp. 679–682 (1988).

[2] J. Yamagishi, C. Veaux, S. King and S. Renals: “Speech synthesis technologies for individuals with vocal disabilities: Voice banking and reconstruction”, Acoustical Science and Technology, **33**, pp. 1–5 (2012).

[3] J. Guerreiro and D. G. calves: “Faster text-to-speeches: Enhancing blind people’s information scanning with faster concurrent speech”, Proc. ASSETS, Lisbon, Portugal, pp. 3–11 (2015).

[4] A. W. Brack: “Speech synthesis for educational technology”, Proc. SLaTE, Farmington, U.S.A., pp. 104–107 (2007).

[5] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao and T.-Y. Liu: “Fastspeech 2: Fast and high-quality end-to-end text to speech”, arXiv, **2006.04558**, (2021).

[6] N. Hojo, Y. Ijima and H. Mizuno: “Dnn-based speech synthesis using speaker codes”, IEICE Transactions on Information and Systems, **E101.D**, 2, pp. 462–472 (2018).

[7] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel and P. Ouellet: “Front-end factor analysis for speaker verification”, IEEE Transactions on Audio, Speech, and Language Processing, **19**, 4, pp. 788–798 (2011).

[8] G. Heigold, I. Moreno, S. Bengio and N. Shazeer: “End-to-end text-dependent speaker verification”, Proc. ICASSP, IEEE Press, pp. 5115–5119 (2016).

[9] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, z. Chen, P. Nguyen, R. Pang, I. Lopez Moreno and Y. Wu: “Transfer learning

from speaker verification to multispeaker text-to-speech synthesis", Advances in Neural Information Processing Systems (Eds. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett), Vol. 31, Curran Associates, Inc. (2018).

[10] G. Hinton: "Reducing the dimensionality of data with neural networks", Science, **313**, pp. 504–507 (2006).

[11] K. Udagawa, Y. Saito and S. Hiroshi: "人間の知覚評価フィードバックによる音声合成の話者適応", Proceedings of the auditory research meeting, **51**, 6, pp. 297–302 (2021).

[12] 前川一高："Corpus of spontaneous japanese: Its design and evaluation", Proc. SSPR, Tokyo, Japan, pp. 7–12 (2003).

[13] S. Takamichi, R. Sonobe, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji and H. Saruwatari: "Jsut and jvs: Free japanese voice corpora for accelerating speech synthesis research", Acoustical Science and Technology, **41**, 5, pp. 761–768 (2020).

[14] S. Hochreiter and J. Schmidhuber: "Long short-term memory", Neural Comput., **9**, 8, pp. 1735–1780 (1997).

[15] X. Glorot, A. Bordes and Y. Bengio: "Deep sparse rectifier neural networks", AISTATS.

[16] A. L. Maas, A. Y. Hannun and A. Y. Ng: "Rectifier nonlinearities improve neural network acoustic models", in ICML Workshop on Deep Learning for Audio, Speech and Language Processing (2013).

[17] D. P. Kingma and J. Ba: "Adam: A method for stochastic optimization" (2014).

[18] Y. Saito, S. Takamichi and H. Saruwatari: "Perceptual-similarity-aware deep speaker representation learning for multi-speaker generative modeling", IEEE/ACM Transactions on Audio, Speech, and Language Processing, **29**, pp. 1033–1048 (2021).