2024年度後期　修士論文

深層学習による口唇音声変換に関する研究

A Study on the Conversion from Lip-Video to Speech Using a Deep Learning Technique

2025年月日

九州大学芸術工学府音響設計コース

2DS23095M

南 汰翼

MINAMI Taisuke

研究指導教員　鏑木 時彦　教授

概要

目 次

1　序論

1.1　背景

音声は基本的なコミュニケーションの手段であり、人と人とのコミュニケーションの場面において、重要な役割を果たしている。音声は、肺からの呼気流による声帯の振動が音源波を生成し、声道特性に伴ったフィルタリングと口唇からの放射特性に従って生成される。これより、音声の生成には音源を作り出す声帯やその制御のための喉頭、舌や口唇といった調音器官の働きが重要となる。しかし、癌などの重い病気で喉頭を摘出した場合、音源波を生成することができなくなるため、これまで通り発声を行うことが不可能になってしまう。このようなコミュニケーション機能の喪失に対し、現在でも電気式人工喉頭や食道発声、シャント発声といった代用音声手法が存在する。電気式人工喉頭では、専用の発振器を顎下に当てて振動を加えることにより、それを音源とした発声を行う。発振器を用意すれば容易に発声することが可能であるが、生成される音声のピッチが発振器の振動に依存してしまうため、抑揚のない単調な音声になってしまう。食道発声では、まず口や鼻から食道内に空気を取り込み、その空気を逆流させることで食道入口部の粘膜を振動させることによって発声する。電気式人工喉頭と違って道具を必要とせず、ピッチも本人が調節できるが、その習得に長期間の訓練を要する。シャント発声では、手術によって気管と食道を繋ぐ管を設ける。これにより、息を吐き出す際に設けられた喉の穴を手で塞ぐことによって肺からの呼気流が食道に流れる。そのため、食道発声と同様に粘膜の振動を音源とし、発声することが可能となる。習得は容易であり、比較的自然に話すことが可能となるが、設けられた管を交換するための定期的な手術が必要となる。このように、現在用いられている代用音声手法にはそれぞれデメリットが存在する。

そのため、本研究ではビデオカメラで撮影した口唇の動きから音声合成を行うことによる、新たな代用音声手法を検討する。本来、音声は声帯の振動や声道の形状に依存して生成されるものであり、口唇の動きのみから音声波形を直接推定することは困難である。そこで、近年画像や自然言語処理、音声といった分野において成果を上げている深層学習を使用し、データ駆動型の方法で口唇の動きと音声の間の関係を学習することで推定を行う。これにより、従来の代用音声手法よりも自然性の高い音声を、訓練や定期的な手術の必要なく提供することを目指す。

これまでの動画音声合成は英語が中心に検討が進んでおり、近年では YouTube 上のデータを収集、処理することによって構築した大規模データセット [1, 2] を用いることで、大規模で表現力の高いモデルが構築可能となっている。これにより、従来行われてきた教師あり学習のみならず、動画と音声の関係性を自己教師あり学習（Self Supervised Learning; SSL）によって学習し、そのモデルを動画音声合成や、動画からテキストを推定する Visual Speech Recognition（VSR）に FineTuning するアプローチが提案され、その有効性が示されている。自己教師あり学習モデルにもいくつかの種類があり、近年多くの研究で応用例のある AVHuBERT [3] は、動画・音声の入力領域においてマスクされた区間の予測と、予測対象の更新を繰り返して学習を進めていくモデルである。予測対象の更新は 5 回行われ、1 回目は音声波形から計算される MFCC をクラスタリングした結果を利用するが、2 回目以降はモデルの中間特徴量をクラスタリング

した結果を新たな予測対象に設定する。更新のたびに再度モデルをランダム初期化して再学習するが、その予測対象の複雑さが増していくことによって学習を促進するようなメカニズムとなっている。また、これに類似した VATLM [4] は、動画と音声のみならずテキストも加えた学習によって、精度改善を達成した。その他、Student と Teacher という二つのネットワークを利用し、Teacher から出力される特徴量を Student がマスクされた入力から予測することによって学習を進める RAVEn [5] や AV-data2vec [6]、RAVEn の改善版として提案された BRAVEn [7] など、多くのモデルが提案されている。

近年の動画音声合成や VSR では、こういった SSL モデルを動画からの特徴抽出器として活用しつつ、さらなる工夫によって精度改善を達成している。動画音声合成について、 [8] では予測対象として従来用いられてきたメルスペクトログラムに加え、テキストを予測するマルチタスク学習手法を提案した。損失においては上記の二つに加え、予測したメルスペクトログラムを事前学習済みの ASR（Automatic Speech Recognition）モデルに入力して得られる特徴表現も採用した。音声波形はメルスペクトログラムに対して Griffin-Lim アルゴリズムを適用することで獲得しており、従来のメルスペクトログラムのみを損失とする手法に対して客観評価指標における改善を達成した。これに続き、 [9] では前述した手法がテキストアノテーションされたデータのみにしか用いることができないという課題を解消するため、テキストと同様に言語的な情報を持つと考えられている、音声 SSL モデルの HuBERT[10] から得られた離散特徴量（HuBERT 離散特徴量とする）を利用する手法を提案した。また、予測されたメルスペクトログラムと HuBERT 離散特徴量の両方を入力とする Multi-input Vocoder、Multi-input Vocoder の学習時にメルスペクトログラムにノイズをかけるデータ拡張を合わせて提案し、客観評価と主観評価の両方で改善を達成した。加えて、ここでは AVHuBERT の転移学習についても合わせて検討が行われ、これによってさらに性能を改善できることを示した。手法 [9] に関連して、上記のようなマルチタスク学習手法以外にも、HuBERT 離散特徴量や HuBERT 連続特徴量（離散化しない場合を指す）を音声波形までの中間特徴量として扱う手法は提案されている。例えば、 [11] ではメルスペクトログラムの推定を行わず、HuBERT 離散特徴量のみを推定して音声波形に変換する手法が提案された。 [9] では離散化におけるクラスタ数を 200 にしていたのに対して、 [11] ではクラスタ数を 2000 と大きく取っている点で実装が異なっている。メルスペクトログラムを省略する分情報圧縮の程度を軽減することで、音声波形への変換に十分な情報を保持する目的があると考えられる。また、[9] や [11] では AVHuBERT を直接動画音声合成に FineTuning していた一方で、 [12] では AVHuBERT を VSR によって FineTuning し、その後重みを固定した上で特徴抽出器として利用するアプローチを提案している。動画音声合成モデルは、VSR で FineTuning した AVHuBERT から得られる動画特徴量を入力とし、HuBERT 特徴量を予測するネットワークを導入して、HuBERT 特徴量のみから音声波形に変換するボコーダを利用することで構築される。ここでは HuBERT 特徴量として連続値および離散値の両方が検討され、連続値を用いる場合の方が客観評価指標が改善することを報告している。検討された離散値のクラスタ数が 100 であったため、[11] の結果と合わせると、HuBERT 離散特徴量のみで音声波形に変換するアプローチを取るのであれば、クラスタ数を十分大きく取る必要があ

ることが予想される。

一方 VSR について、 [13] では音声認識を利用して言語情報を格納したメモリを用意し、メモリと動画特徴量の間でアテンションをとることによって、ネットワーク内部で言語情報との関連を考慮する構成を提案した。また、 [14] では AVHuBERT が動画あるいは音声のどちらを入力とした場合でもクロスモーダルな特徴量を返すことに着目し、音声認識デコーダに組み合わせる AVHuBERT の Few-shot Learning、Zero-shot Learning による転移学習を検討した。加えて、同様に音声認識デコーダを転移学習するアプローチであるが、事前学習済みモデルの重みを固定し、動画特徴量から音声認識モデルの中間特徴量を予測するネットワークのみを新たに学習することで、両者を合併するようなアプローチ [15] も提案されている。さらに、静止画像と音声から動画を合成するネットワークを構築し、音声認識用のデータセットを用いて VSR の学習データを大量に合成するデータ拡張手法 [16] や、事前学習済みの音声認識モデルによって教師なしデータにラベリングを行うデータ拡張手法 [17]、10 万時間分の教師ありデータを新たに増強した研究 [18] など、大規模な学習データを確保することで精度改善を達成した例も報告されている。

上記の研究は英語データを用いたものであったが、VSR においては英語以外の言語に焦点を当てた研究や、多言語対応モデルの構築も検討が進んでいる。 [19] では RAVEn を利用し、英語に加えてスペイン語、イタリア語、ポルトガル語など計 6 種類の言語が含まれるデータセット [20, 21, 22] を用いて多言語モデルの構築を検討した。結果として、教師ありデータの少ない英語以外の言語に対する、多言語モデルの有効性が明らかとなった。また、 [23] では英語データで学習された AVHuBERT を用いつつ、特定の言語ごとに構築した音声認識モデルのデコーダを転移学習することで、特定言語ごとにモデルを構築するアプローチを提案した。さらに、[24] では音声認識モデルである Whisper を利用し、教師なしデータへのラベリングによるデータ拡張を行うことで、上記二つのアプローチを超える精度を達成した。

本研究では、近年の主流とも言える英語大規模データセットを用いた実験は計算機のスペックの都合上難しく、世界的に見て日本語での動画音声合成の検討例が少ないことも考慮して、文献 [25, 26] で収録された日本語データを用いて研究を行うこととした。英語データと比較して小規模なデータである分性能に課題を抱えたが、予備実験として英語データで学習された AVHuBERT の FineTuning を検討したところ、スクラッチで構築したモデルと比較して、より高い精度を示すことが明らかとなった。これは、英語データを用いた事前学習済みモデルの多言語対応を検討した先行研究の傾向にも一致する結果であり、日本語においても同様に有効なモデルだと考えられる。しかしながら、それでも依然として合成音声の品質は低く、自然音声に迫る合成音は実現されていないことが課題である。

1.2 目的

本研究の目的は、動画音声合成によって得られる合成音声の品質を向上させることである。近年高い精度を達成した手法 [9] では、AVHuBERT の利用および、メルスペクトログラムと音

声 SSL 離散特徴量を利用したマルチタスク学習が採用されている。その他にも近年高い精度を達成したモデルは存在 [11, 12, 27] するが、手法 [9] が採用しているマルチタスク学習の有効性は、テキストを用いた先行研究 [8] でも同様に示されている。これより、このアプローチが現状特に有効そうだと判断し、本研究においてはこの手法をベースラインとして、さらなる改善を狙う形で研究を進めることとした。この手法では、動画を入力としてメルスペクトログラムと音声 SSL 離散特徴量を推定し、これら両方を Multi-input Vocoder に入力することで音声波形へと変換する。しかし、動画と音声の間には、同様の口の動きであっても声道形状の違いによって生じる発話内容の曖昧さや、話者によるパターンの多様さが存在すると考え、推定を動画のみに依存した先行研究の手法ではこういった側面への対処が難しいと考えた。これに対して本研究では、音声 SSL モデルである HuBERT を利用した動画音声合成モデルを提案し、合成音声の推定残差を HuBERT を利用した後処理によって軽減することで、合成音声の品質改善を狙った。HuBERT は、音声波形を畳み込み層を通すことによってダウンサンプリングしつつ特徴量に変換し、ここでマスクをかけた上で Transformer 層を通す。そして、マスクされたフレームにおける予測対象を推定する、Masked Prediction を行うことで学習する。大規模な音声データを用いてこの自己教師あり学習を行うことで、音声のコンテキスト自体をデータそのものから学習することが可能であり、音声認識において有効性が確認されている。本研究では、大規模日本語音声データで事前学習済みの HuBERT を活用し、動画音声合成モデルにおいて生じる推定残差を、音声自体のコンテキストを考慮する形で補うようなアプローチを検討した。

1.3 本論文の構成

2　音声信号処理

3　深層学習

4　動画音声合成モデルの検討

4.1　音声合成法

提案手法の構築手順は 3 段階に分かれる。ネットワークの概要を図 4.1 に示す。一段階目では、動画を入力として、メルスペクトログラムと HuBERT 離散特徴量、HuBERT 中間特徴量を推定するネットワーク A を学習する（図 4.1 の A）。ここで、HuBERT 離散特徴量は HuBERT Transformer 層から得られる特徴量を k-means 法によってクラスタリングすることで離散化した結果、HuBERT 中間特徴量は HuBERT における畳み込み層出力のことを指す。図 4.2 にこれらの取得位置を示す。第一段階では、AVHuBERT を動画からの特徴抽出に利用した。これにより、動画の空間情報は完全に圧縮され、768 次元の一次元系列となる。その後、事前学習済みの話者識別モデル [28] によって音声波形から得られる 256 次元の話者 Embedding を、各フレームでチャンネル方向に結合する。これによって特徴量は 1024 次元に拡張され、全結合層によって再度 768 次元に圧縮する。その後、畳み込み層と全結合層からなる ConvDecoder（図 4.1 の ConvDecoder）を通すことによって、話者 Embedding を結合した特徴量に対する変換を施した。これにより、特にメルスペクトログラムにおいて話者性が正しく反映されることを狙った。動画の見た目から話者性を判断できる可能性も考えられたが、本研究では念の為入力することとした。ConvDecocer は残差結合を利用したブロック単位で構成され、各ブロックに 2 層の畳み込み層を設けた。各畳み込み層のチャンネル数は 768、カーネルサイズは 3 であり、3 ブロック積み重ねた。最後に全結合層を通し、所望の次元に変換することで予測対象を得た。ネットワーク A の役割は、続くネットワーク B の入力である HuBERT 中間特徴量を提供することである。これに対し、メルスペクトログラムと HuBERT 離散特徴量の推定を同時に行った理由は、先行研究においてマルチタスク学習の有効性が確認されていることを考慮し、ネットワーク A でもマルチタスク学習を採用しておこうと考えたからである。また、こうすることで後に述べるネットワーク C に向けた拡張性も得られたため、これを一貫して用いることとした。

二段階目では、一段階目に学習されたネットワーク A の重みを固定した状態で HuBERT 中間特徴量を推定し、それを入力としてメルスペクトログラムと HuBERT 離散特徴量を推定する、HuBERT Transformer 層を中心としたネットワーク B の学習を行う（図 4.1 の B）。HuBERT Transformer 層出力は AVHuBERT 出力と同じ 768 次元の特徴量となるため、これに対してネットワーク A と同様に話者 Embedding を結合し、ConvDecoder を通すことで予測値を得た。ネットワーク B の役割は、音声波形への変換に必要となるメルスペクトログラムと HuBERT 離散特徴量の予測である。HuBERT Transformer 層の転移学習を検討した狙いについて、HuBERT は自己教師あり学習時、畳み込み層出力にマスクを適用し、Transformer 層を通すことによってマスクされた部分を推定しようとする。これにより、音声の文脈を考慮するのに長けた学習済み重みが、特に Transformer 層で獲得されると仮定した。これに基づき、本研究では HuBERT Transformer 層を動画音声合成に Fine Tuning することにより、動画を入力とした AVHuBERT を中心とするネットワーク A における推定残差を、音声自体の文脈を考慮することによって軽減し、動画から直接推定しきれなかった部分を補うことでの精度改善を狙った。

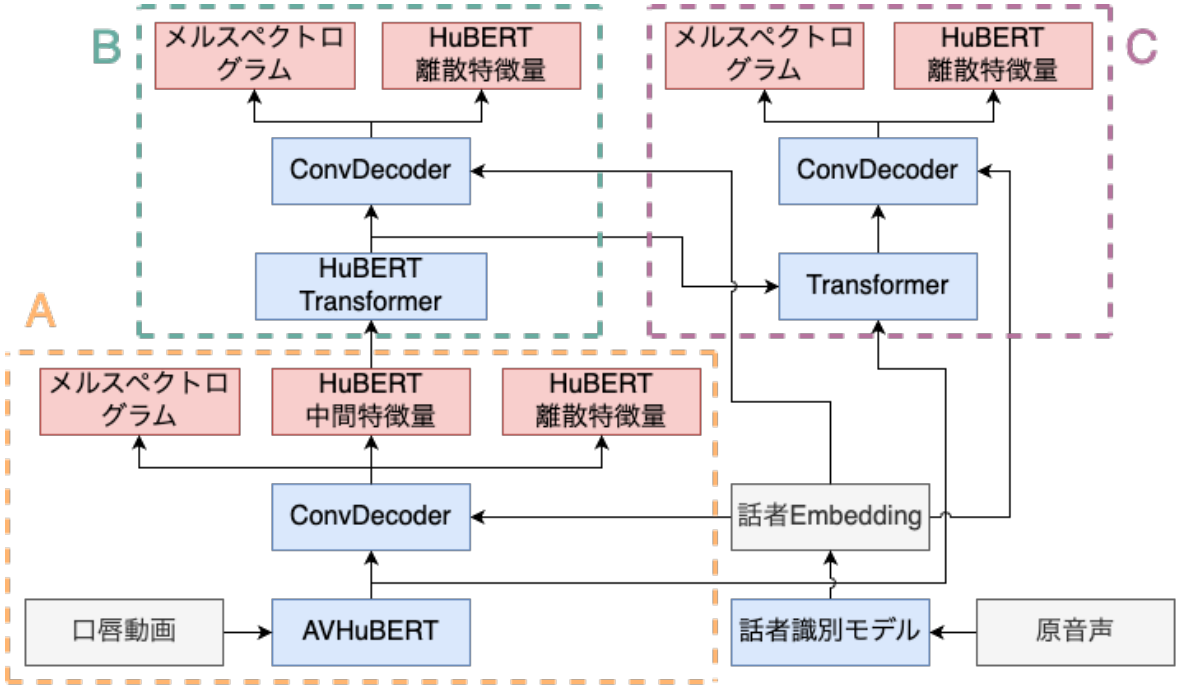


図 4.1: ネットワークの概略図

三段階目では、二段階目までに学習されたネットワーク A とネットワーク B の重みを固定した状態で、AVHuBERT から得られる特徴量と、HuBERT Transformer 層から得られる特徴量の二つを結合し、それらを入力として再びメルスペクトログラムと HuBERT 離散特徴量の予測を行うネットワーク C を学習した（図 4.1 の C）。ネットワーク C では、はじめに前述した二つの特徴量をチャンネル方向に結合することで、1536 次元の入力特徴量を得る。これに対して全結合層を施すことで再度 768 次元に圧縮し、4 層の Transformer 層を通すことで系列全体を考慮した特徴抽出を改めて行った。その後、ネットワーク A,B と同様に話者 Embedding を結合し、ConvDecoder を通すことによって予測値を得た。ここで、ネットワーク C の Transformer 層におけるパラメータについては、AVHuBERT や HuBERT と同様にチャンネル数を 768、ヘッド数は 12 とした。ネットワーク C の役割は、ネットワーク B と同様に音声波形への変換に必要な特徴量の予測である。ここでの狙いについて、まず、AVHuBERT から得られる特徴量と HuBERT Transformer 層から得られる特徴量は、どちらも ConvDecoder への入力となる点で同じである。一方、AVHuBERT は動画を入力、HuBERT Transformer 層は HuBERT 中間特徴量を入力とするため、これら特徴量の元となる入力は異なっている。ここでは、概ね同じ予測対象のために利用される二つの特徴量（ネットワーク A では HuBERT 中間特徴量の予測も行っているため、全く同じではない）が、入力の違いに依存して内部の Self Attention により注意される部分が変化し、何らかの異なった情報を持っている可能性があると仮定した。この仮定に基づき、両方の特徴量を考慮して単一特徴量への依存を解消することで、汎化性能向上による予測精度の改善を狙った。

以上が提案手法の全体像であるが、今回ベースラインとする先行研究 [9] に基づいたマルチタスク学習手法は、本研究におけるネットワーク A で、HuBERT 中間特徴量を推定しないも

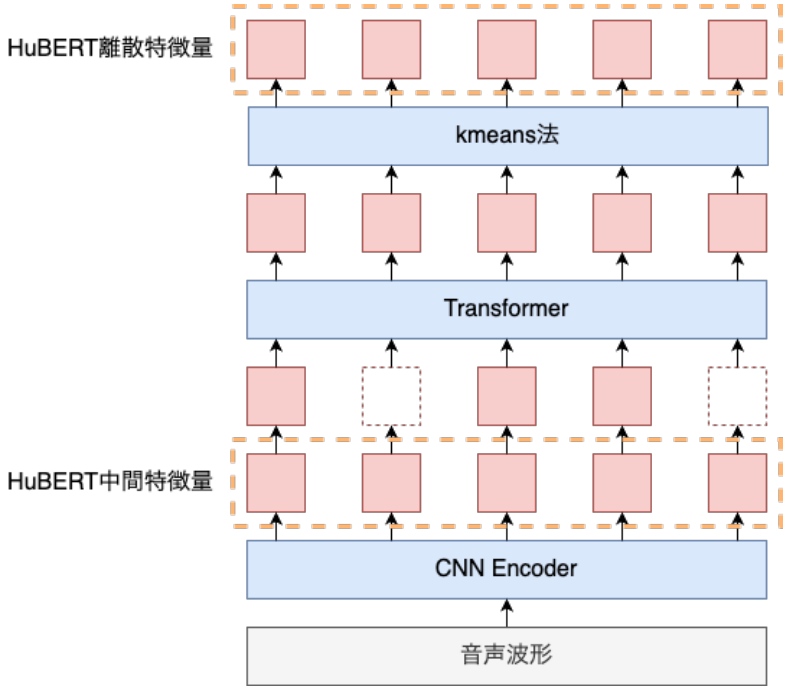


図 4.2: HuBERT 中間特徴量と HuBERT 離散特徴量の概略

のに当たる。AVHuBERT 以降の ConvDecoder について、これは先行研究と異なる構成であるが、これは実験の過程でより良いものを選択した結果である。本実験においてはこれを比較手法全てで一貫して用いることで、比較したい部分には影響が出ないようにしつつ、全体的な性能の底上げを狙った。

以上のモデルにより、動画からメルスペクトログラムと HuBERT 離散特徴量が推定可能となる。その後、先行研究 [9] に基づく Multi-input Vocoder を用い、メルスペクトログラムと HuBERT 離散特徴量を入力として音声波形に変換することで、最終的な合成音声を得た。Multi-input Vocoder は HiFi-GAN [29] をベースとしたモデルであり、音声波形を生成する Generator と、Multi Scale Discriminator および Multi Period Discriminator という二つの Discriminator によって構成される。Generator への入力特徴量について、メルスペクトログラムは 100Hz、HuBERT 離散特徴量は 50Hz となっているが、メルスペクトログラムは隣接したフレームを次元方向に縦積みすることで、100Hz・80 次元の特徴量から 50Hz・160 次元の特徴量に変換して用いた。入力時は全結合層によって 128 次元の特徴量に変換した。一方、HuBERT 離散特徴量はインデックス系列であるが、入力時はインデックスから 128 次元のベクトルに変換した。その後、これらをチャンネル方向に結合することで 256 次元の特徴量を構成し、これを Generator への最終的な入力とした。その後は、HiFi-GAN の公式実装と同様である。

表 4.1: 利用したデータセットの文章数

	学習	検証	テスト
動画音声データセット	1598	200	212
Hi-Fi-Captain	37714	200	200
JVS	10398	1299	1300

4.2 実験方法

4.2.1 利用したデータセット

動画音声データセットには、男女二人ずつから収録した合計 4 人分のデータセット [25, 26] を用いた。これは ATR 音素バランス文 [30] から構成され、全話者共通で A から H セットを学習データ、I セットを検証データ、J セットをテストデータとして利用した。各分割ごとの文章数を表 4.1 に示す。

Multi-input Vocoder の学習に利用する音声データセットには、Hi-Fi-Captain（日本語話者二名分） [31] と JVS（parallel100 と nonpara30） [32] を利用した。ボコーダの学習時にはサンプル全体から 1 秒分をランダムにサンプリングして用いるが、元データには話し声のない無音区間が一定存在しており、これはボコーダの学習に望ましくない。これに対して、無音区間のトリミング（-40 dBFS 未満かつ 500 ms 継続する区間を 100 ms までカット）を適用した。Hi-Fi-Captain は train-parallel および train-non-parallel を学習データ、val を検証データ、eval をテストデータとして分割した。各分割ごとの文章数を表 4.1 に示す。JVS には話者に対して 1 から 100 まで番号が割り振られており、本実験では 1 から 80 番の話者を学習データ、81 番から 90 番の話者を検証データ、91 番から 100 番までの話者をテストデータとした。各分割ごとの文章数を表 4.1 に示す。

4.2.2 データの前処理

動画データは 60 FPS で収録されたものを ffmpeg により 25 FPS に変換して用いた。その後、手法 [33] により動画に対してランドマーク検出を適用した。このランドマークを利用することで口元のみを切り取り、画像サイズを（96, 96）にリサイズした上で、グレースケールに変換した。加えて、画像に対する正規化および標準化を適用した。全体として、今回は事前学習済みの AVHuBERT の転移学習を行うため、そこでの前処理に合わせている。学習時は、ランダムクロップ、左右反転、Time Masking（一時停止）をデータ拡張として適用した。ランダムクロップは、（96, 96）で与えられる画像から（88, 88）をランダムに切り取る処理である。検証およびテスト時は、必ず画像中央を切り取るよう実装した。左右反転はランダムクロップ後に適用しており、50%の確率で左右が反転されるよう実装した。Time Masking は、連続する画像の時間平均値を利用することによって、一時停止させるような効果を与えるデータ拡張手法で

ある。動画 1 秒あたり 0 から 0.5 秒の間でランダムに停止区間を定め、その区間における動画の時間方向平均値を計算し、区間内のすべてのフレームをこの平均値で置換した。

音声データは 48 kHz で収録されたものを 16 kHz にダウンサンプリングして用いた。それから、窓長 25 ms のハニング窓を用いて、シフト幅 10 ms で STFT を適用することでフレームレート 100 Hz のスペクトログラムに変換した。さらに、振幅スペクトログラムに対して 80 次のメルフィルタバンクを適用し、メルスペクトログラムを得た上で対数スケールに変換した。話者 Embedding の取得には事前学習済みの話者識別モデル [28] を利用し、学習データから 100 個ランダムサンプリングして計算した平均値を利用した。

HuBERT は、HuggingFace に公開されている ReazonSpeech というデータセットによって学習されたモデル [34, 35] を利用した。ReazonSpeech は約 19000 時間の日本語音声からなるデータセットであり、日本語音声のコンテキストを大量のデータから学習したモデルである。今回用いるデータセットが日本語であることから、本研究の検討対象としては日本語音声に関する事前知識を豊富に有するモデルが適していると考え、このモデルを選択した。動画からの予測対象となる HuBERT 離散特徴量は、k-means 法によるクラスタリング（クラスタ数 100）を HuBERT Transformer 層の 8 層目出力に適用することで得た。8 層目を選択した理由は、HuBERT のレイヤーごとの特徴量について、音素の One-hot ベクトルおよび単語の One-hot ベクトルとの相関を、Canonical Correlation Analysis（CCA）によって調べた先行研究 [36] より、8 層目出力がそのどちらとも相関が高く言語的な情報に近いと判断したからである。k-means 法の学習には動画音声データセットにおける学習用データを利用し、これによって動画音声データおよび、Multi-input Vocoder の学習に用いる外部の音声データについてクラスタリングを適用した。

4.2.3 学習方法

一段階目について、損失関数はメルスペクトログラムの MAE Loss L_{mel} と HuBERT 離散特徴量の Cross Entropy Loss L_{ssl^d} 、HuBERT 中間特徴量の MAE Loss L_{ssl^i} の重み付け和とした。それぞれの重み係数を $\lambda_{mel}, \lambda_{ssl^d}, \lambda_{ssl^i}$ とすると、

$$L = \lambda_{mel} * L_{mel} + \lambda_{ssl^d} * L_{ssl^d} + \lambda_{ssl^i} * L_{ssl^i} \tag{4.1}$$

となる。第一段階では $\lambda_{mel} = 1.0, \lambda_{ssl^i} = 1.0$ と固定し、λ_{ssl^d} のみ 0.0001 から 1.0 まで 10 倍刻みで 5 段階試してチューニングを行った。最適化手法には AdamW [37] を利用し、$\beta_1 = 0.9$、$\beta_2 = 0.98$、$\lambda = 0.01$ とした。学習率は 1.0×10^{-3} から開始し、Warmup Scheduler によってその値を変化させた。バッチサイズはメモリの都合上 4 としたが、学習の安定のため Gradient Accumulation によって各イテレーションにおける勾配を累積させ、8 イテレーションに一回重みを更新するようにした。そのため、実質的にはバッチサイズ 32 となる。モデルに入力する動画の秒数は 10 秒を上限とし、それを超える場合はランダムにトリミング、それに満たない場合はゼロパディングした。勾配のノルムは 3.0 を上限としてクリッピングすることで、過度に大きくなることを防止した。最大エポック数は 50 とし、10 エポック連続して検証データに対す

る損失が小さくならない場合には、学習を中断するようにした（Early Stopping）。また、学習終了時には検証データに対する損失が最も小さかったエポックにおけるチェックポイントを保存し、これをテストデータに対する評価に用いた。

第二段階について、損失関数はメルスペクトログラムの MAE Loss と HuBERT 離散特徴量の Cross Entropy Loss の重み付け和とした。これは式 (4.1) において、$\lambda_{mel} = 1.0, \lambda_{ssl^i} = 0.0$ と固定した場合に相当する。第一段階と同様に、λ_{ssl^d} のみ 0.0001 から 1.0 まで 10 倍刻みで 5 段階試してチューニングを行った。最適化手法には AdamW を利用し、$\beta_1 = 0.9$、$\beta_2 = 0.98$、$\lambda = 0.01$ とした。学習率は 5.0×10^{-4} から開始し、Warmup Scheduler によってその値を変化させた。学習率について第一段階と異なるが、これは値を半減させることによって学習を安定させることができたためである。その他のパラメータは、第一段階における値と同じである。

第三段階について、学習率のみ第一段階と同様に 1.0×10^{-3} としたが、それ以外は第二段階と全く同じである。

Multi-input Vocoder の学習について、学習は動画音声データセットではなく、外部データである Hi-Fi-Captain と JVS を用いた。はじめに Hi-Fi-Captain のみを用いて学習させ、その後学習済みモデルを JVS によって再学習した。Hi-Fi-Captain は男女一人ずつの文章数が豊富なデータセットであるため、高品質なモデルを構築可能であった。しかし、学習できる話者数が少ない分、学習外話者に対する合成音声の品質が低かった。そのため、一人当たりの文章数は 100 文章程度と少ないながらも、100 人分の話者からなる JVS を利用して再学習することによって、学習外話者に対する合成音声の品質を向上させた。最適化手法には AdamW を利用し、$\beta_1 = 0.8$、$\beta_2 = 0.99$、$\lambda = 0.01$ とした。学習率は 2.0×10^{-4} から開始し、指数関数的にその値を変化させた。バッチサイズは 16 とし、ここでは Gradient Accumulation は利用しなかった。モデルへの入力は 1 秒を上限とし、それを超える場合はランダムにトリミング、それに満たない場合はゼロパディングした。勾配のノルムは 3.0 を上限としてクリッピングすることで、過度に大きくなることを防止した。最大エポック数は 30 とし、ここでは Early Stopping は適用しなかった。また、学習終了時には検証データに対する損失（メルスペクトログラムに対する L1 Loss）が最も小さかったエポックにおけるチェックポイントを保存し、これをテストデータに対する評価に用いた。また、Multi-input Vocoder の提案された先行研究 [9] においては、学習時にあえてメルスペクトログラムにノイズをかけることによって、合成音声に対する汎化性能を向上させる学習方法が提案されている。本研究では、動画から推定されるメルスペクトログラムと HuBERT 離散特徴量の推定精度向上に焦点を当てたため、Multi-input Vocoder の学習は原音声から計算される特徴量そのもので行い、ボコーダ自体の汎化性能向上による精度改善は追求しなかった。

実装に用いた深層学習ライブラリは PyTorch および PyTorch Lightning である。GPU には NVIDIA RTX A4000 を利用し、計算の高速化のため Automatic Mixed Precision を適用した。

4.2.4 比較手法

比較手法は、以下の五つである。

1. メルスペクトログラムと HuBERT 離散特徴量のマルチタスク学習手法（ベースライン）
2. 提案手法のネットワーク B で、ランダム初期化した HuBERT Transformer 層を用いる手法
3. 提案手法のネットワーク C で、ランダム初期化した HuBERT Transformer 層を用いる手法
4. 提案手法のネットワーク B で、事前学習済みの HuBERT Transformer 層を用いる手法
5. 提案手法のネットワーク C で、事前学習済みの HuBERT Transformer 層を用いる手法

手法 1 が先行研究において有効性が確認された手法であり、今回の実験においてベースラインとなる。これに対する改善案として、手法 2 から手法 5 が提案手法である。手法 2 および手法 3 は、今回期待する HuBERT の事前学習済み重みを初期値とした転移学習が本当に有効であるか検証する目的で、あえてランダム初期化した HuBERT Transformer を用いる場合である。

4.2.5 客観評価

合成音声の客観評価では、二種類の指標を用いた。一つ目は、音声認識の結果から算出した Word Error Rate（WER）である。Whisper [38] を用いて音声認識を行い、出力される漢字仮名交じり文に対して MeCab を用いて分かち書きを行った上で、jiwer というライブラリを用いて算出した。Whisper は Large モデルを利用し、MeCab の辞書には unidic を利用した。WER の値は 0%から 100%であり、この値が低いほど音声認識の誤りが少ないため、より聞き取りやすい音声であると判断した。二つ目は、話者 Embedding から計算したコサイン類似度である。話者 Embedding の計算は、モデルへの入力値を計算したものと同様の話者識別モデルを利用し、対象音声と原音声のペアでコサイン類似度を計算した。今回構築するモデルは 4 人の話者に対応するモデルとなるため、原音声に似た声質の合成音声が得られているかをこの指標で評価した。値は 0 から 1 であり、高いほど原音声と類似した合成音声だと判断できる。

4.2.6 主観評価

合成音声の主観評価では、音声の明瞭性と類似性の二点を評価した。今回はクラウドワークスというクラウドソーシングサービスおよび、自作の実験用 Web サイトを利用してオンラインで実験を実施した。被験者の条件は、日本語母語話者であること、聴覚に異常がないこと、イヤホンあるいはヘッドホンを用いて静かな環境で実験を実施可能であることとした。被験者の方に行っていただいた項目は、以下の五つである。

1. アンケート
2. 練習試行（明瞭性）
3. 本番試行（明瞭性）
4. 練習試行（類似性）
5. 本番試行（類似性）

一つ目のアンケートでは、被験者についての基本的な統計を取ることを目的として、性別・年齢・実験に利用した音響機器について回答してもらった。性別は、男性、女性、無回答の三

つからの選択式とした。年齢は被験者の方に直接数値を入力してもらう形式とした。実験に使用した音響機器は、イヤホン、ヘッドホンの二つからの選択式とした。

二つ目の練習試行（明瞭性）および三つ目の本番試行（明瞭性）では、音声の明瞭性の評価を実施した。初めに練習試行を行っていただくことで実験内容を把握してもらい、その後本番施行を行っていただく流れとした。ここで、練習施行は何度でも実施可能とし、本番試行は一回のみ実施可能とした。評価項目について、明瞭性は「話者の意図した発話内容をその通り聞き取ることができるか」を評価するものとした。実際の評価プロセスは以下の三段階で構成した。

1. 音声サンプルのみを聞いてもらい、その音声の発話内容を聞き取ってもらう。
2. 発話内容を聞き取ることができた、あるいはこれ以上聞き取ることはできないと判断したら、本来の発話内容を確認してもらう。
3. 聴取者が想定していた発話内容と本来の発話内容を照らし合わせ、音声の聞き取りやすさを 5 段階評価してもらう。

5 段階評価の回答項目は以下のようにした。

1. 全く聞き取れなかった
2. ほとんど聞き取れなかった
3. ある程度聞き取れた
4. ほとんど聞き取れた
5. 完全に聞き取れた

実験に利用した音声サンプルについて、練習試行では検証データ、本番試行ではテストデータを用いた。評価対象とした音声の種類は、4.2.4 節における五つの手法と、原音声、分析合成を加えた 7 種類である。被験者ごとの評価サンプルの割り当て方法について、初めに評価に用いる文章を比較手法の総数である 7 つのグループにランダムに分割した。具体的には、練習試行では検証データである ATR 音素バランス文の I セットから 7 種類、本番試行ではテストデータである ATR 音素バランス文の J セットのすべて、すなわち 53 種類の文章を 7 つの文章グループにランダムに分割した。次に、各文章グループに対して 7 種類の手法から一つを割り当てることで、文章一つ一つに対する手法の割り当てを行った。残る話者の決定については、ランダムに割り当てるようにした。文章グループに対して割り当てる手法は、被験者ごとにずらすよう実装した。この過程を表 4.2 に示す。これは、7 つの文章グループに対し、割り当てる手法が一つずつずれていく過程を表している。これにより、文章と手法の組み合わせを効率よく網羅できる [39]。またこの選択方法により、各話者はすべての文章を一回ずつ評価する機会が与えられ、その中で各手法がなるべく均等な回数含まれることとなる。同じ発話内容の音声を二回以上提示しないことで、聴取者の集中力を保つことを狙った。また、各手法をなるべく均等な回数提示するようにした理由は、手法の比較が主観評価の最終的な目的であったため、各被験者がすべての手法を評価する機会を与えたかったからである。加えて、サンプル選択の過程では、以前に選択された回数をカウントしておくことで、サンプルの選択にランダム性を持たせつつ、すべてのサンプルが等しい回数評価されるようにした。例えば、一回選択されたサンプ

表 4.2: 主観評価実験のサンプル選択における文章グループと手法の対応関係

	文章グループインデックス						
	1	2	3	4	5	6	7
手法インデックス	1	2	3	4	5	6	7
	2	3	4	5	6	7	1
	3	4	5	6	7	1	2
	4	5	6	7	1	2	3
	5	6	7	1	2	3	4
	6	7	1	2	3	4	5
	7	1	2	3	4	5	6

ルと未選択のサンプルが存在する場合、一回選択されたサンプルは選択の候補から除外する。これにより、未選択のサンプルのみを対象としたランダムサンプリングを行うことで、最終的な評価回数が等しくなるよう実装した。本実験では手法が七種類、話者が四人存在するため、28 回の実験によって、手法・話者・文章のすべての組み合わせが一回ずつ評価される、すなわち全サンプルが一回ずつ評価されることになる。

四つ目の練習試行（類似性）および五つ目の本番試行（類似性）では、評価対象の音声と同一話者の原音声の類似性の評価を実施した。ここでも初めに練習試行を行っていただくことで実験内容を把握してもらい、その後本番施行を行っていただく流れとした。ここで、練習施行は何度でも実施可能とし、本番試行は一回のみ実施可能とした。評価項目について、類似性は「評価対象の音声が同一話者の原音声とどれくらい似ているか」を評価するものとした。実際の評価プロセスは以下の二段階で構成した。

1. 評価対象の音声と原音声を聞き比べてもらう。
2. 評価対象の音声が原音声にどれくらい似ていたかを五段階評価してもらう。

5 段階評価の回答項目は以下のようにした。

1. 全く似ていなかった
2. あまり似ていなかった
3. やや似ていた
4. かなり似ていた
5. 同じ話者に聞こえた

実験に利用した音声サンプルおよび、被験者ごとの評価サンプルの割り当て方法は明瞭性の評価実験と同様である。ただし、類似性評価においては同一話者の原音声を発話内容についてランダムに選択し、評価対象となるサンプルとペアで提示できるようにした。評価時は、明瞭性評価と同様に音声サンプルを何度でも聞けるようにしたが、発話文章については提示しなかった。なぜなら、類似性評価では評価が発話文章に依存しないからである。実際、評価サンプルのペアとなる原音声サンプルは発話文章をランダムに選択しているため、一致する場合も異な

る場合も存在する。

また、オンラインでの評価は効率よく数多くの方に評価していただけるという点でメリットがあるが、オフラインでの評価と比較して実験環境を制御することが難しく、評価品質が低下する恐れがある。これに対して、本実験では先行研究 [40] を参考に、評価サンプル中にダミー音声を混入させることで対策を講じた。ダミー音声は本研究で得られた合成音声とは無関係に、gTTS というライブラリを用いて生成したサンプルである。具体例として、明瞭性評価では

> これはダミー音声です。明瞭性は「3: ある程度聞き取れた」を選択してください。

のような発話内容の音声を、類似性評価では

> これはダミー音声です。類似性は「1: 全く同じ話者には聞こえなかった」を選択してください。

のような発話内容の音声を提示した。この時、その音声自体の明瞭性や類似性とは無関係に、必ずこの音声によって指定された評価値を選択するよう説明を与えた。本番試行においてダミー音声で指定された評価値を誤って選んだ場合は、すべての回答を無効にする旨を被験者に伝えた。実際、実験終了後にはそのようにデータを処理した。

被験者数および各手法の評価回数に関して、先行研究 [41] では主観評価実験の結果に対する統計処理について、そこで用いる被験者数や手法ごとの評価回数を変数とし、実験条件に対してどれほどの被験者数とサンプル数が必要そうであるかを検討している。今回はこの研究を参考にしつつ、オンラインで実験を実施するのであれば総被験者数が 100 人以上、各手法に対する総評価回数が 200 回以上となることが望ましいと判断した。前述した各被験者に対するサンプルの選択方法により、28 回の実験によって全てのサンプルが一回ずつ評価される。これを 1 セットとすると、セットあたり被験者数は 28 人、各手法に対する評価回数は 212 回となる。従って、今回は 4 セット行うことで、総被験者数 112 人、各手法に対しての総評価回数が 848 回となるようにした。実験は 30 分程度で終わると見積もって、一人当たりの報酬は 500 円とした。

4.3 結果

4.3.1 客観評価

まず、損失関数 (4.1) の重み係数 λ_{ssl^d} を変化させた時の、客観評価指標の全テストデータに渡る平均値を表 4.3 に示す。各手法ごとに 0.0001 から 1.0 まで 10 倍刻みで 5 段階検討し、各手法の客観指標ごとに最も優れた値を下線で示している。最良エポックは検証データに対する損失が最小となったエポックであり、テストデータの合成にはこのエポックにおけるチェックポイントを利用した。また、L_{mel}、L_{ssl^d}、L は最良エポックにおける検証データに対する損失の平均値である。また、これ以降の比較のために、最適だと考えられる λ_{ssl^d} の値を選択しており、選択された行を太字で表している。

手法 1 では、λ_{ssl^d} の値が 0.0001 の時に WER が最も低く、0.01 の時に話者類似度が最も高くなった。現状 WER の高さが特に課題であり、話者類似度は 0.004 とわずかな違いでもあるため、今回は WER が最小であることを優先して 0.0001 が最適であると判断した。λ_{ssl^d} の値による評価指標の変化について、WER は λ_{ssl^d} の値を大きくするのに伴って単調に増加していることがわかる。一方、話者類似度は λ_{ssl^d} の値が 0.1 以上となった時に、0.01 以下であった場合と比較して顕著に低下することがわかる。次に、図 4.3 に手法 1 における学習曲線の結果を示す。横軸がエポック数、縦軸が損失の値を表す。損失の値は各エポックにおける平均値である。実線は検証データに対する損失、点線は学習データに対する損失を表しており、線の色は λ_{ssl^d} の違いを表す。また、丸いマーカーは表 4.4 に示した最良エポック時における損失の値を表す。学習曲線より、λ_{ssl^d} の値を変化させることによって、特に L_{ssl^d} の傾向が変化していることがわかる。具体的には、λ_{ssl^d} の値を 0.0001 から 1.0 へと増加させるのに伴って、学習初期における損失の下がり方が急峻になっており、達する最小値自体が小さくなっていることがわかる。また、λ_{ssl^d} の値が 0.1 以上の場合、検証データに対する L_{ssl^d} は早いうちから増加傾向に転じている。これに伴い、今回は検証データに対する L の値を監視し、Early Stopping の適用と最良エポックの決定を行なったため、L_{mel} が下がり切らない状態で学習が中断される結果となった。客観評価指標において、特に λ_{ssl^d} の値が 0.1 以上となるとき、0.01 以下の場合と比較して話者類似度の低下や、WER の上昇傾向が見られていたが、学習曲線の挙動より、L_{mel} を下げきれなくなっていたことが原因として考えられる。最適な λ_{ssl^d} の値は客観評価指標から 0.0001 としたが、0.01 以下ではそれと概ね同程度の品質であったことを考えると、手法 1 では検証データに対する L_{mel} の値を十分小さくすることのできる λ_{ssl^d} が適していると考えられる。

手法 2 では、λ_{ssl^d} の値が 0.1 の時に WER が最も低く、0.0001 の時に話者類似度が最も高くなった。ここでは λ_{ssl^d} の値が 0.1 の時に、ベースラインで選択された最適なケースと比較して WER が 9.1%低下しており、話者類似度についても 0.003 高くなっていることから、0.1 が最適だと判断した。λ_{ssl^d} の値による評価指標の変化について、λ_{ssl^d} を 0.1 以上とすることで、0.01 以下の場合と比較して WER が低下する傾向が見られた。しかし、1.0 まで大きくすると 0.1 の場合よりも WER が大きくなっており、単調に減少していないこともわかる。話者類似度については、λ_{ssl^d} の値を 0.1 としたときに、0.01 以下の場合と比較して値が少し低下し、さらに 1.0 まで大きくすることで 0.1 以下の場合と比較して顕著に低下していることがわかる。次に、図 4.4 に手法 2 における学習曲線の結果を示す。手法 1 と同様に、λ_{ssl^d} の値を 0.0001 から 1.0 へと増加させるのに伴って、学習初期における L_{ssl^d} の下がり方が急峻になっており、達する最小値自体が小さくなっていることがわかる。また、λ_{ssl^d} が 1.0 の場合に、L_{mel} を下げきれなくなる傾向が見られる。加えて、手法 2 では λ_{ssl^d} が 0.1 の場合における L_{ssl^d} の増加が緩やかであり、L_{mel} も十分下げられていることがわかる（ただし、表 4.3 の L_{ssl^d} より、手法 1 の λ_{ssl^d} が 0.1 の場合と比較して、損失の値自体は大きい）。さらに、λ_{ssl^d} が 0.1 の場合、L_{mel} が達する最小値自体が、λ_{ssl^d} が 0.01 以下の場合と比較して小さくなっていることがわかる（具体的な値は表 4.3 の L_{mel} の値を参照）。これは手法 1 では見られなかった新たな傾向であった。最適な λ_{ssl^d} の値は客観評価指標から 0.1 としたが、学習曲線の挙動より、この時他の値の場合と比較して L_{mel}

と L_{ssl^d} の両方をバランスよく下げられていたことがわかった。よって、手法 2 においては L_{mel} と L_{ssl^d} の両方をバランスよく下げられるような、程よい大きさの重みが適していると考えられる。

手法 3 における最適値の選択理由は手法 2 と同様であり、λ_{ssl^d} の値による評価指標の変化についても同様であった。次に、図 4.5 に手法 3 における学習曲線の結果を示す。傾向は手法 2 と概ね同様であるが、手法 3 では λ_{ssl^d} の値が 0.1 の場合における L_{ssl^d} の増加が早く、学習が早期に中断されている点は異なる。しかし、この時 L_{mel} も十分下げられており、L_{mel} が達する最小値自体が、λ_{ssl^d} が 0.01 以下の場合と比較して小さくなっていることがわかる（具体的な値は表 4.3 の L_{mel} の値を参照）。以上より、細かな違いはあるものの手法 3 は手法 2 と同様の挙動を示し、λ_{ssl^d} は L_{mel} と L_{ssl^d} の両方をバランスよく下げられるような、程よい大きさの重みが適していると考えられる。

手法 4 及び手法 5 についても、最適値の選択理由は手法 2 と同様である。しかし、これらは λ_{ssl^d} の値を 0.1 としたときの話者類似度の低下の度合いが手法 2、手法 3 と比較して大きい点で、傾向が異なっていた。次に、図 4.6 に手法 4、図 4.7 に手法 5 における学習曲線の結果を示す。傾向については、L_{ssl^d} の増加の程度について細かな違いはあるが、概ね手法 2 と同様であり、L_{mel} と L_{ssl^d} の両方をバランスよく下げられるような、程よい大きさの重みが適していると考えられる。また、手法 4 と手法 5 において、λ_{ssl^d} が 0.1 の場合における話者類似度は、手法 2 および手法 3 と比較して低くなっていたが、L_{mel} や L_{ssl^d} の値からは一貫した傾向が見られない。これより、平均値からこの違いを分析することは難しいが、実際の細かな誤差の出方が異なっていて、これが影響を与えているのではないかと考える。

次に、最適なチューニングをした場合における、手法ごとの客観評価指標の全テストデータに渡る平均値を表 4.4 に示す。分析合成は、原音声から計算した特徴量を入力として、Multi-input Vocoder で逆変換した合成音声であり、本実験下において合成音声により達成され得る上限値を表す。手法 1 から手法 5 については、表 4.3 において太字としたもの、すなわち最適なチューニングだと判断されたものを選択している。また、分析合成、原音声を除いた合成音声（手法 1 から手法 5）の中で、最も優れた値を下線で示している。これより、WER、話者類似度ともに手法 3 が最も優れていることが分かる。特に、今回のベースラインである手法 1 と比較すると、WER が 9.5%低下し、話者類似度は 0.01 高くなっていることから、提案手法がベースラインよりもより聞き取りやすく、話者性を反映した音声を合成できたと考えられる。また、HuBERT の事前学習済み重みを初期値とすることの効果について、手法 2 と手法 4 を比較すると、手法 2 の方が WER が 1.2%低く、話者類似度が 0.017 高いことがわかる。よって、本研究では HuBERT の事前学習済み重みを初期値とした転移学習が有効である可能性を検討したが、むしろ事前学習済み重みを初期値とせず、ランダム初期化したモデルの方が優れていたと考えられる。これは仮説に反した結果であったが、事前学習済み重みによって与えられる初期値が、より良い局所解への収束には繋がらなかったことが原因だと考えられる。一方、アンサンブル手法の有効性について、手法 2 と手法 3 を比較すると、アンサンブル手法を導入することによって WER が 0.4%低下し、話者類似度が 0.007 高くなっていることから、わずかではあるが改善している

ことがわかる。しかし、手法 4 と手法 5 を比較すると、WER は変化せず、話者類似度は 0.017 低下していることから、特に話者類似度の観点で悪化したことがわかる。よって、アンサンブル手法は必ずしも改善につながるわけではなく、改善するとしても顕著な変化はもたらさないと考えられる。

次に、最適なチューニングをした場合における手法ごとに、発話文章ごとの WER の比較を行った結果を図 4.8 に示す。ここで、縦軸は発話文章を表し、横軸はベースラインである手法 1 とその他手法の間で WER の差を計算し、発話文章ごとに平均した値を表す。負の値を取っているとき、手法 1 に対してより低い WER を達成したと解釈できる。図より、表 4.4 における平均値のみを見れば、手法 2 から手法 5 の全てが手法 1 に対してより低い WER を達成していたが、発話文章ごとに見れば、手法 1 に対して WER が高くなっているサンプルもあることがわかる。例えば、「ATR503_j11」では、手法 2 から手法 5 の全てがベースラインよりも 20% 前後高い WER となっていることがわかる。また、全体的な傾向として手法ごとに WER はばらけており、いかなる場合においても最良となるようなモデルは構築できていないことがわかる。これより、現状のいかなるモデルも発話文章に対する汎化性能が不十分であり、さらなる改善が必要だと考えられる。

次に、話者ごとの WER の比較を行った結果を図 4.9 に示す。ここで縦軸は話者を表し、横軸はベースラインである手法 1 とその他手法の間で WER の差を計算し、話者ごとに平均した値を表す。負の値を取っているとき、手法 1 に対してより低い WER を達成したと解釈できる。これより、手法 2 から手法 5 は平均的にいかなる話者に対しても手法 1 より低い WER を達成していることがわかるが、一方でその改善の度合いは話者によって異なることがわかる。特に、「F02_kablab」はその他三人の話者と比較して改善の度合いが小さい。また、例えば「F01_kablab」では手法 2 と手法 3 がより有効である一方、「F02_kablab」では手法 4 と手法 5 がより有効となっており、有効な手法が話者によって異なることもわかる。これより、話者に対する性能の依存があると考えられ、さらなる汎化性能の向上が必要だと言える。

次に、話者ごとの話者類似度の比較を行った結果を図 4.10 に示す。ここで縦軸は話者を表し、横軸はベースラインである手法 1 とその他手法の間で話者類似度の差を計算し、話者ごとに平均した値を表す。正の値を取っているとき、手法 1 に対してより高い話者類似度を達成したと解釈できる。これを見ると、手法 4 と手法 5 はいかなる話者に対しても話者類似度を悪化させたことがわかる。手法 4 と手法 5 は事前学習済み重みを初期値としたモデルであったため、この影響が出ている可能性がある。今回検討した HuBERT は音声の言語情報を考慮する点で強みがあり、実際音声認識ではその転移学習の有効性が確認されている。しかし、話者性を考慮するという点で特に有効でなく、話者性を軽視するような局所解への収束に繋がった可能性が考えられる。ただ、表 4.3 より手法 4 と手法 5 においても λ_{ssl^d} が 0.01 以下であれば、比較的高い話者類似度を達成していたため、損失の重み付けにも依存していると言える。また、手法 3 はいかなる話者に対しても類似度を向上させており、この点で優れていたと言える。手法 2 では「F02_kablab」のみ話者類似度が悪化しており、手法 3 と比較して話者に対する汎化性能が低かったと考えられる。

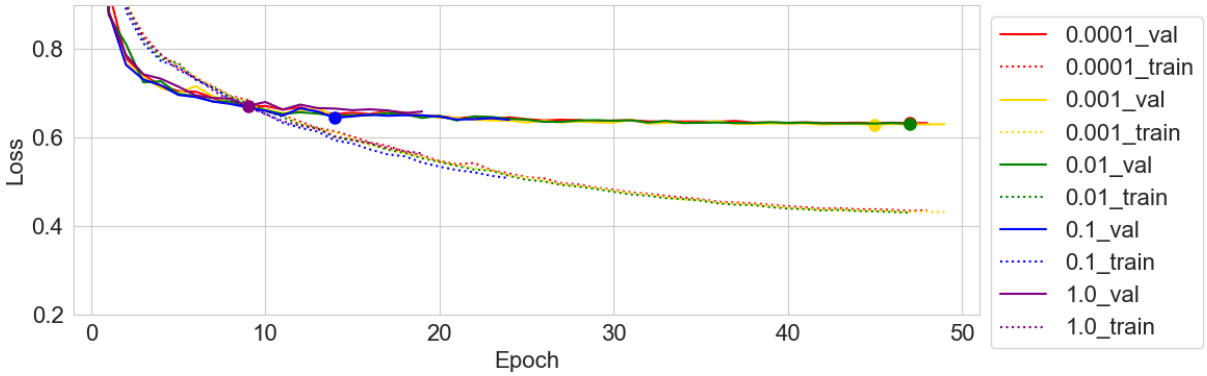
以上のことから、提案手法である手法2から手法5はいずれも手法1に対して平均的にWERを低下させ、特に手法2及び手法3は話者類似度についても若干の改善を達成したことから、提案手法によるベースラインからの改善が達成できたと考える。また、HuBERTの事前学習済み重みを用いることは仮説に反して有効ではなく、アンサンブル手法についてもその効果は顕著なものではなかった。さらに、特に手法2から手法5は損失関数の重み係数であるλ_{ssl^d}による性能の変化が著しく、ここで最適値を選択できなければベースラインである手法1から改善しないこともわかった。よって、提案手法におけるベースラインからの改善に寄与した可能性のあるポイントは、以下でまとめられる。

1. HuBERT中間特徴量を入力とし、メルスペクトログラムとHuBERT離散特徴量を推定するネットワークを導入したこと
2. 上記のネットワーク構造をHuBERT Transformerとしたこと
3. ネットワークをランダム初期化した上で学習させたこと
4. 最適なλ_{ssl^d}の値を発見できたこと

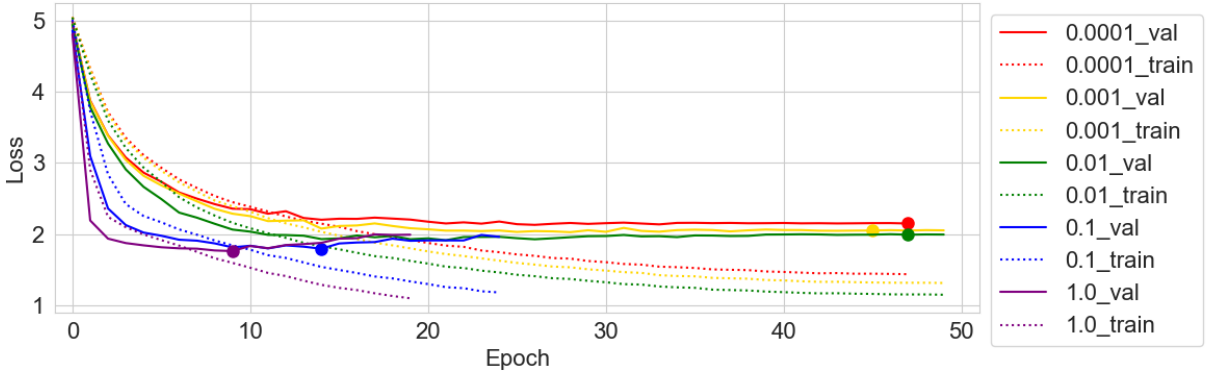
本実験からは、HuBERT Transformerをランダム初期化する必要性と、λ_{ssl^d}の値のチューニングを行う必要性が明らかとなった。しかし、入力特徴量やネットワーク構造についてはHuBERTを用いることを前提とした実験条件であったことから、検討できていない。実験結果から、もはやHuBERTに依存する必要性は無くなっているため、入力がHuBERT中間特徴量でなくても良いし、ネットワークも任意に選択できる。入力については、例えば最終予測値であるメルスペクトログラムやHuBERT離散特徴量を採用しても実装は可能であるが、これによって性能が変化するのであれば、HuBERT中間特徴量が良い入力特徴量であると考えられる。ネットワークについては、Transformer自体が多くの場合有効であるため現状で適切なものとなっている可能性もあるが、検討の余地はある。こういった点が、今後の検討課題として挙げられる。

表 4.3: 損失関数の重み係数 λ_{ssl^d} による客観評価指標の比較

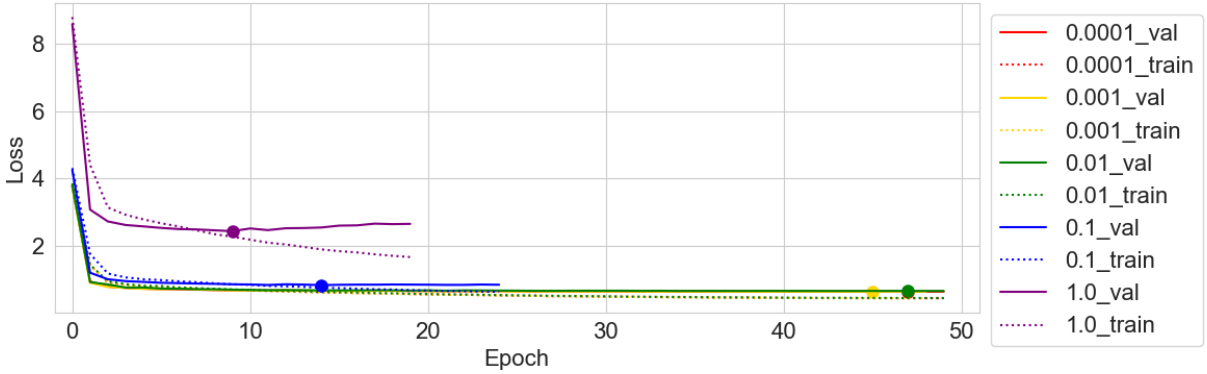
手法	λ_{ssl^d}	WER [%]	話者類似度	最良エポック	L_{mel}	L_{ssl^d}	L
1	**0.0001**	**55.3**	**0.841**	**47**	**0.632**	**2.147**	**0.633**
1	0.001	55.6	0.841	45	0.629	2.049	0.631
1	0.01	56.7	0.845	47	0.631	1.990	0.651
1	0.1	57.7	0.764	14	0.645	1.793	0.824
1	1.0	62.1	0.699	9	0.672	1.760	2.431
2	0.0001	58.9	0.860	46	0.630	2.521	0.630
2	0.001	58.3	0.857	45	0.631	2.382	0.634
2	0.01	56.7	0.859	48	0.631	2.104	0.652
2	**0.1**	**46.2**	**0.844**	**32**	**0.620**	**1.942**	**0.814**
2	1.0	50.3	0.706	9	0.651	1.713	2.364
3	0.0001	58.8	0.860	22	0.632	2.584	0.632
3	0.001	58.1	0.854	22	0.633	2.448	0.636
3	0.01	56.5	0.865	45	0.630	2.077	0.651
3	**0.1**	**45.8**	**0.851**	**18**	**0.622**	**2.041**	**0.826**
3	1.0	48.6	0.755	14	0.634	1.735	2.370
4	0.0001	58.2	0.849	20	0.632	2.593	0.632
4	0.001	57.4	0.847	24	0.633	2.496	0.635
4	0.01	56.0	0.853	30	0.630	2.106	0.651
4	**0.1**	**47.4**	**0.827**	**17**	**0.622**	**1.929**	**0.815**
4	1.0	48.4	0.720	10	0.648	1.746	2.394
5	0.0001	57.8	0.857	22	0.632	2.547	0.632
5	0.001	57.5	0.853	27	0.635	2.449	0.637
5	0.01	58.1	0.865	44	0.632	2.100	0.653
5	**0.1**	**47.4**	**0.810**	**6**	**0.633**	**1.981**	**0.831**
5	1.0	47.7	0.748	16	0.637	1.786	2.423



(a) メルスペクトログラムの MAE Loss（式 (4.1) の L_{mel}）

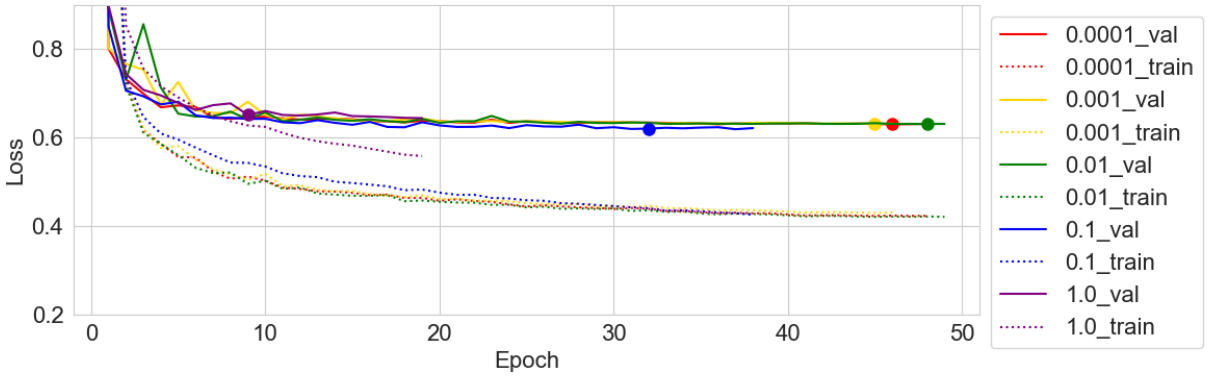


(b) HuBERT 離散特徴量の Cross Entropy Loss（式 (4.1) の L_{ssl^d}）

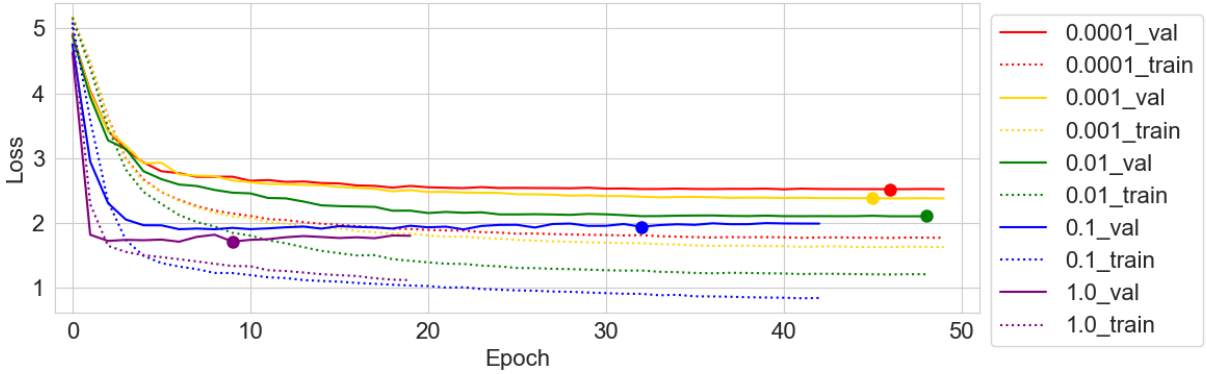


(c) 損失の合計値（式 (4.1) の L）

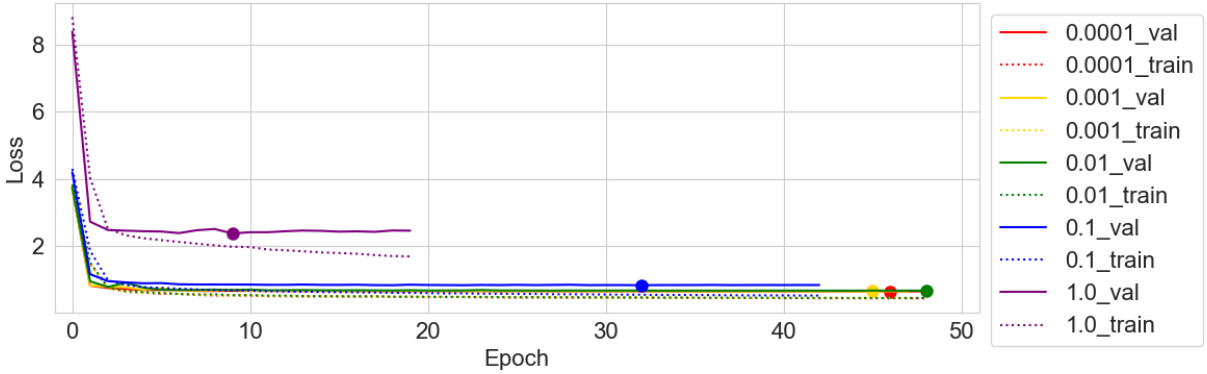
図 4.3: 手法 1 における学習曲線



(a) メルスペクトログラムの MAE Loss（式 (4.1) の L_{mel}）

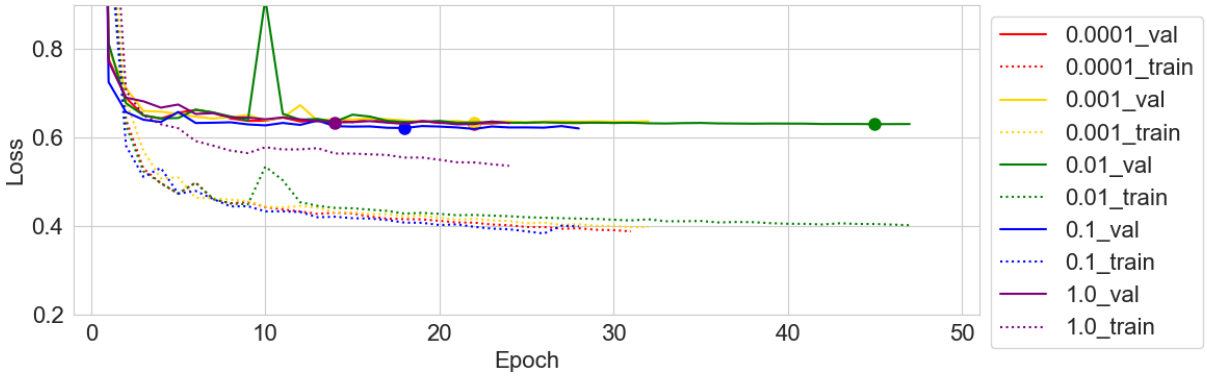


(b) HuBERT 離散特徴量の Cross Entropy Loss（式 (4.1) の L_{ssl^d}）

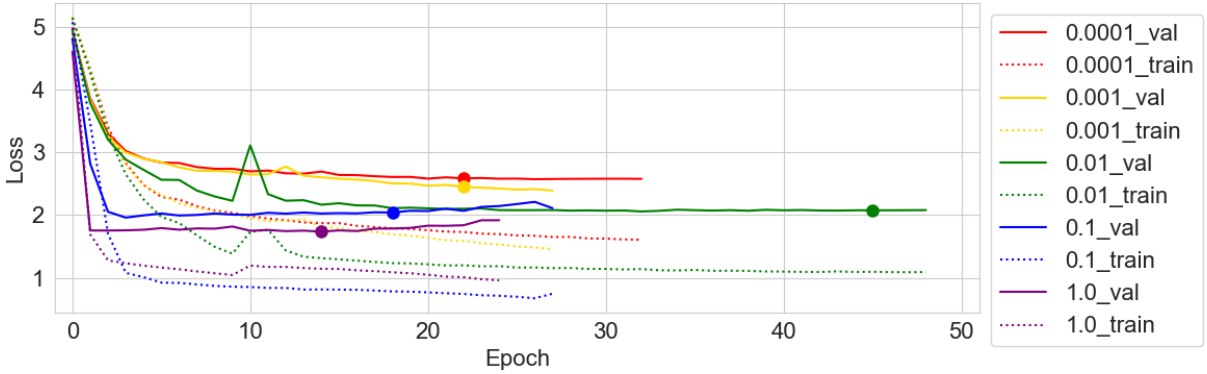


(c) 損失の合計値（式 (4.1) の L）

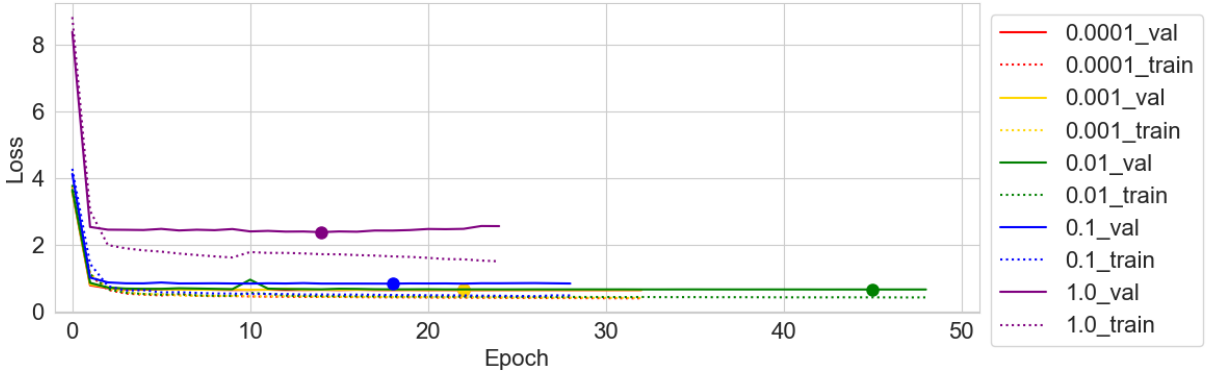
図 4.4: 手法 2 における学習曲線



(a) メルスペクトログラムの MAE Loss（式 (4.1) の L_{mel}）

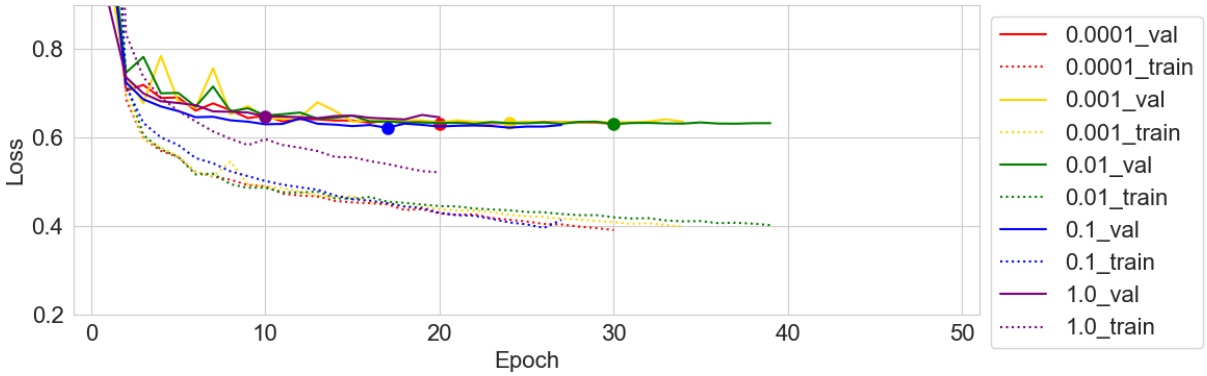


(b) HuBERT 離散特徴量の Cross Entropy Loss（式 (4.1) の L_{ssl^d}）

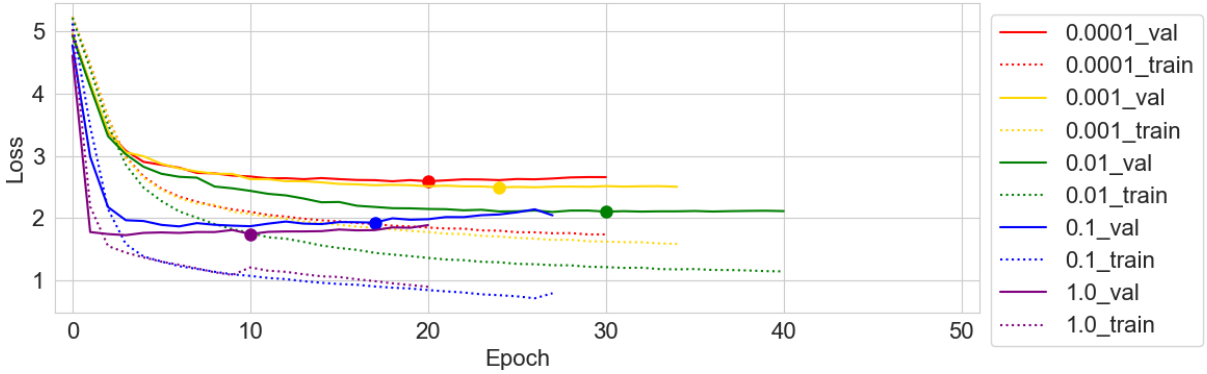


(c) 損失の合計値（式 (4.1) の L）

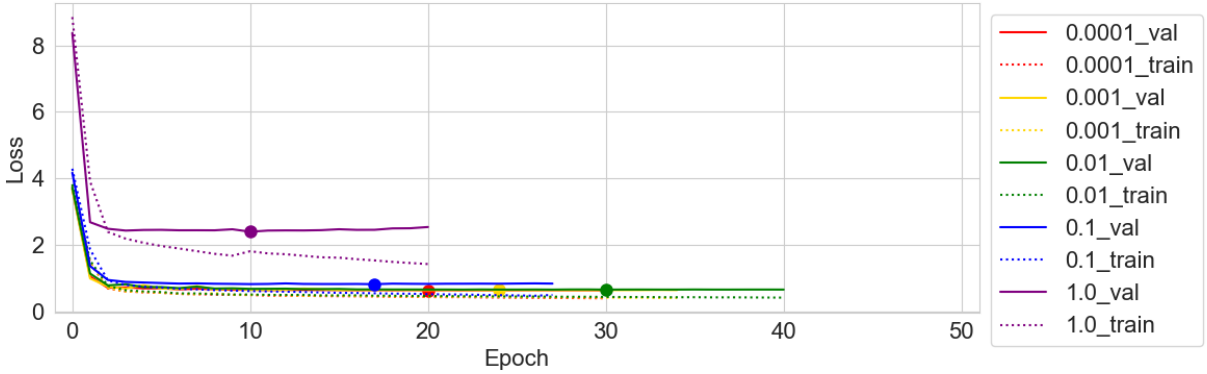
図 4.5: 手法 3 における学習曲線



(a) メルスペクトログラムの MAE Loss（式 (4.1) の L_{mel}）

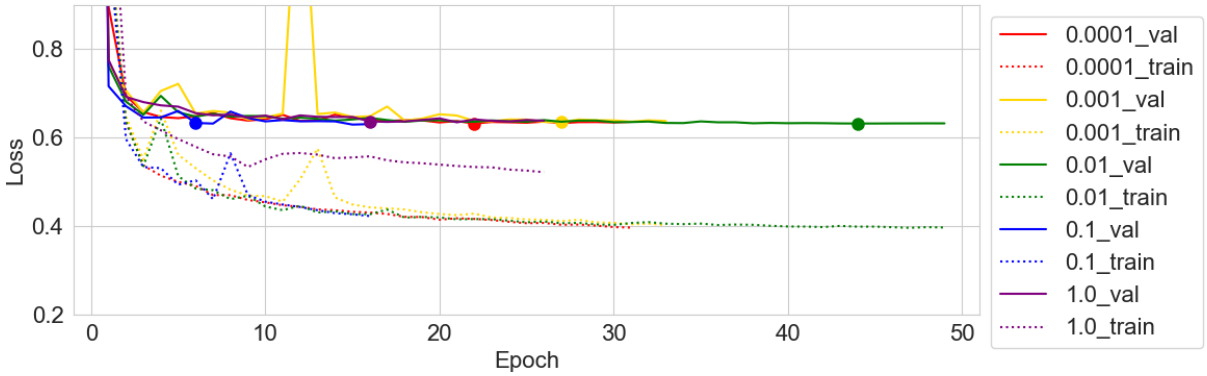


(b) HuBERT 離散特徴量の Cross Entropy Loss（式 (4.1) の L_{ssl^d}）

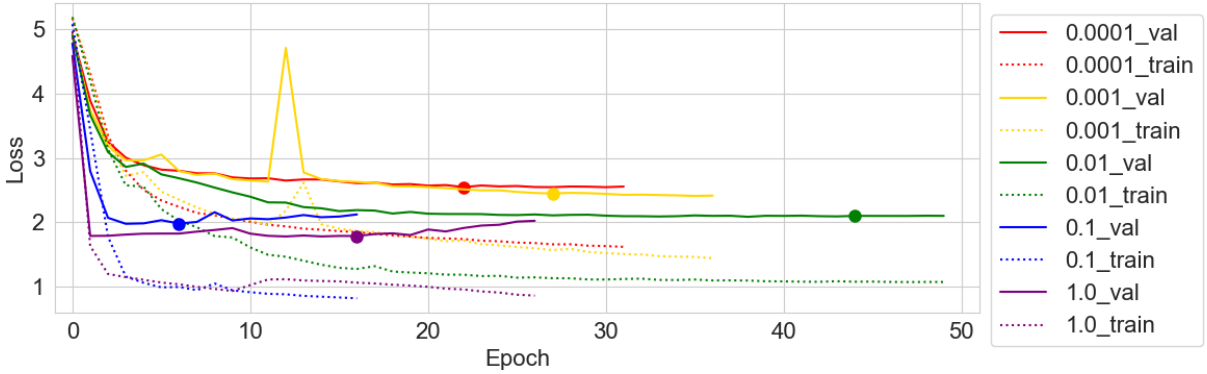


(c) 損失の合計値（式 (4.1) の L）

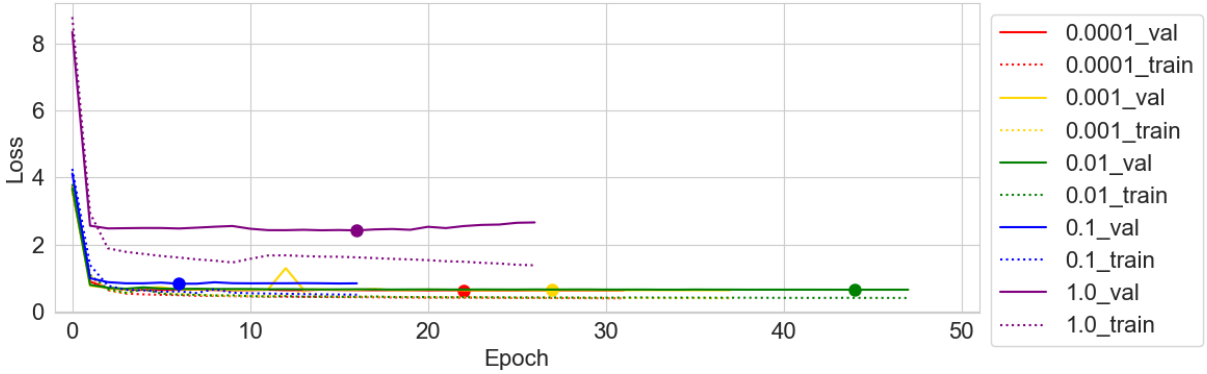
図 4.6: 手法 4 における学習曲線



(a) メルスペクトログラムの MAE Loss（式 (4.1) の L_{mel}）



(b) HuBERT 離散特徴量の Cross Entropy Loss（式 (4.1) の $L_{sslᵈ}$）



(c) 損失の合計値（式 (4.1) の L）

図 4.7: 手法 5 における学習曲線

表 4.4: 最適なチューニングをした場合における手法ごとの比較

手法	詳細	WER [%]	話者類似度
1	ベースライン	55.3	0.841
2	ランダム初期化 Transformer	46.2	0.844
3	ランダム初期化 Transformer + アンサンブル	<u>45.8</u>	<u>0.851</u>
4	事前学習あり Transformer	47.4	0.827
5	事前学習あり Transformer + アンサンブル	47.4	0.810
6	分析合成	4.8	0.944
7	原音声	4.5	1.000

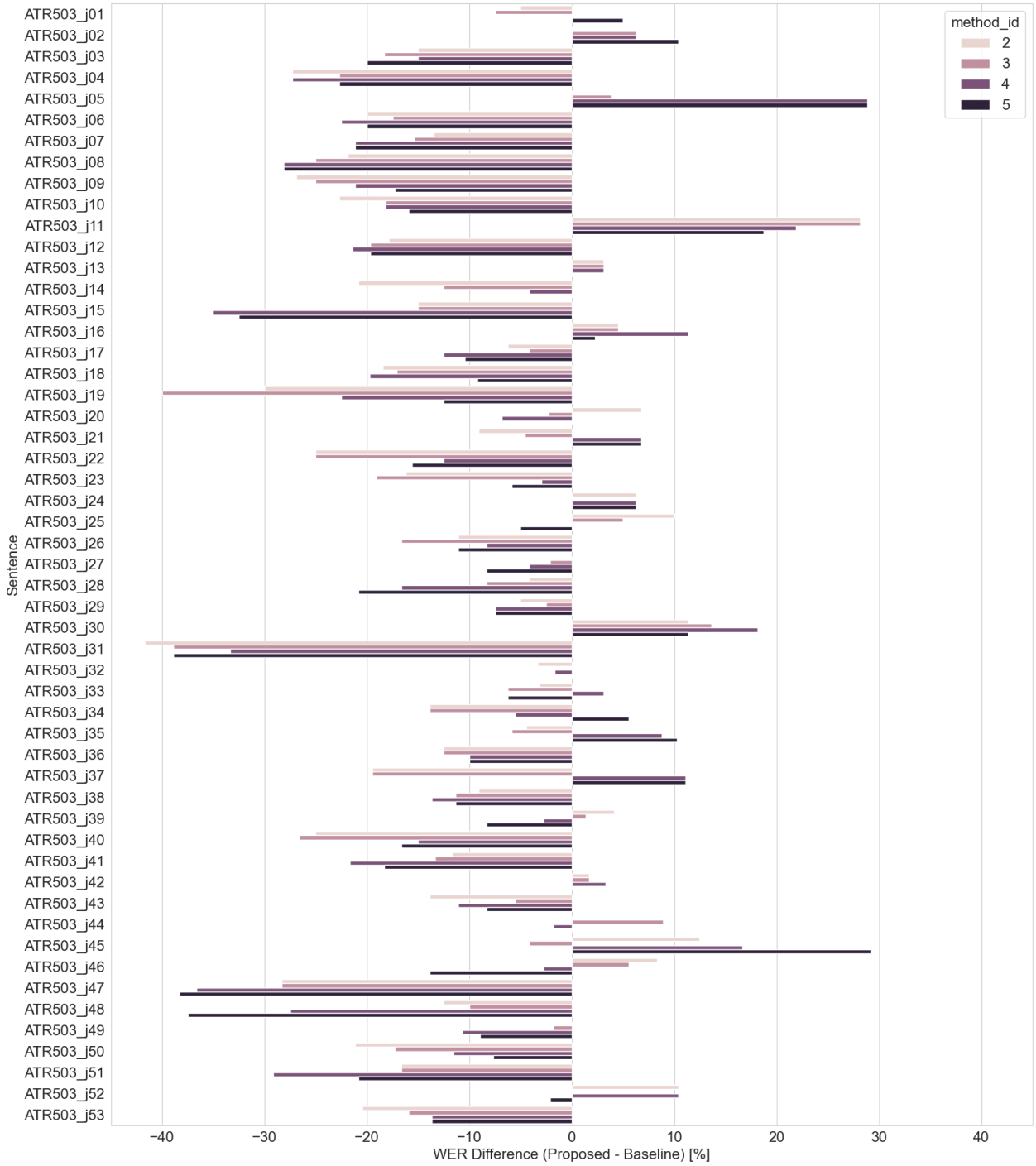


図 4.8: テストデータにおける発話文章ごとの WER の比較

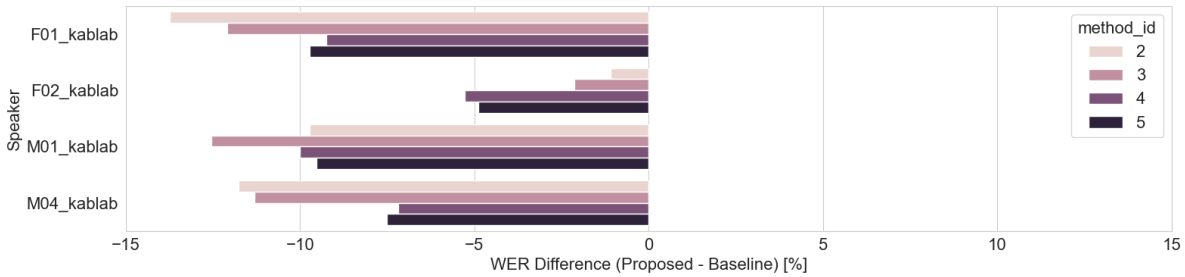


図 4.9: テストデータにおける話者ごとの WER の比較

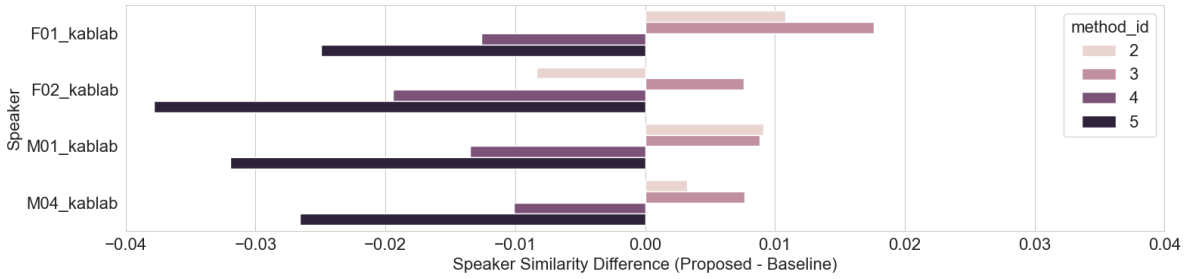


図 4.10: テストデータにおける話者ごとの話者類似度の比較

4.3.2 主観評価

4.4　まとめ

5　結論

謝辞

参考文献

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Lrs3-ted: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496*, 2018.

[2] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018.

[3] Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. Learning audio-visual speech representation by masked multimodal cluster prediction. *arXiv preprint arXiv:2201.02184*, 2022.

[4] Qiushi Zhu, Long Zhou, Ziqiang Zhang, Shujie Liu, Binxing Jiao, Jie Zhang, Lirong Dai, Daxin Jiang, Jinyu Li, and Furu Wei. Vatlm: Visual-audio-text pre-training with unified masked prediction for speech representation learning. *IEEE Transactions on Multimedia*, 2023.

[5] Alexandros Haliassos, Pingchuan Ma, Rodrigo Mira, Stavros Petridis, and Maja Pantic. Jointly learning visual and auditory speech representations from raw data. *arXiv preprint arXiv:2212.06246*, 2022.

[6] Jiachen Lian, Alexei Baevski, Wei-Ning Hsu, and Michael Auli. Av-data2vec: Self-supervised learning of audio-visual speech representations with contextualized target representations. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. IEEE, 2023.

[7] Alexandros Haliassos, Andreas Zinonos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. Braven: Improving self-supervised pre-training for visual and auditory speech recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11431–11435. IEEE, 2024.

[8] Minsu Kim, Joanna Hong, and Yong Man Ro. Lip-to-speech synthesis in the wild with multi-task learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

[9] Jeongsoo Choi, Minsu Kim, and Yong Man Ro. Intelligible lip-to-speech synthesis with speech units. *arXiv preprint arXiv:2305.19603*, 2023.

[10] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, Vol. 29, pp. 3451–3460, 2021.

[11] Wei-Ning Hsu, Tal Remez, Bowen Shi, Jacob Donley, and Yossi Adi. Revise: Self-supervised speech resynthesis with visual input for universal and generalized speech regeneration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18795–18805, 2023.

[12] Neha Sahipjohn, Neil Shah, Vishal Tambrahalli, and Vineet Gandhi. Robustl2s: Speaker-specific lip-to-speech synthesis exploiting self-supervised representations. In *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1492–1499. IEEE, 2023.

[13] Jeong Hun Yeo, Minsu Kim, Jeongsoo Choi, Dae Hoe Kim, and Yong Man Ro. Akvsr: Audio knowledge empowered visual speech recognition by compressing audio knowledge of a pretrained model. *IEEE Transactions on Multimedia*, 2024.

[14] Xize Cheng, Tao Jin, Linjun Li, Wang Lin, Xinyu Duan, and Zhou Zhao. Opensr: Open-modality speech recognition via maintaining multi-modality alignment. *arXiv preprint arXiv:2306.06410*, 2023.

[15] Yasser Abdelaziz Dahou Djilali, Sanath Narayan, Haithem Boussaid, Ebtessam Almazrouei, and Merouane Debbah. Lip2vec: Efficient and robust visual speech recognition via latent-to-latent visual to audio representation mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13790–13801, 2023.

[16] Xubo Liu, Egor Lakomkin, Konstantinos Vougioukas, Pingchuan Ma, Honglie Chen, Ruiming Xie, Morrie Doulaty, Niko Moritz, Jachym Kolar, Stavros Petridis, et al. Synthvsr: Scaling up visual speech recognition with synthetic supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18806–18815, 2023.

[17] Pingchuan Ma, Alexandros Haliassos, Adriana Fernandez-Lopez, Honglie Chen, Stavros Petridis, and Maja Pantic. Auto-avsr: Audio-visual speech recognition with automatic labels. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

[18] Oscar Chang, Hank Liao, Dmitriy Serdyuk, Ankit Shahy, and Olivier Siohan. Conformer is all you need for visual speech recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 10136–10140. IEEE, 2024.

[19] Andreas Zinonos, Alexandros Haliassos, Pingchuan Ma, Stavros Petridis, and Maja Pantic. Learning cross-lingual visual speech representations. In *ICASSP 2023-2023 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE, 2023.

[20] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018.

[21] Elizabeth Salesky, Matthew Wiesner, Jacob Bremerman, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W Oard, and Matt Post. The multilingual tedx corpus for speech recognition and translation. *arXiv preprint arXiv:2102.01757*, 2021.

[22] Ya Zhao, Rui Xu, and Mingli Song. A cascade sequence-to-sequence model for chinese mandarin lip reading. In *Proceedings of the 1st ACM International Conference on Multimedia in Asia*, pp. 1–6, 2019.

[23] Minsu Kim, Jeong Hun Yeo, Jeongsoo Choi, and Yong Man Ro. Lip reading for low-resource languages by learning and combining general speech knowledge and language-specific knowledge. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15359–15371, 2023.

[24] Jeong Hun Yeo, Minsu Kim, Shinji Watanabe, and Yong Man Ro. Visual speech recognition for low-resource languages with automatic labels from whisper model. *arXiv preprint arXiv:2309.08535*, 2023.

[25] 田口史郎. ”深層学習を用いたデータ駆動型調音・音声間変換に関する研究”. 九州大学大学院芸術工学府芸術工学専攻 博士論文, 2021.

[26] 江崎蓮. ”深層学習を用いた口唇動画・音声変換に関する調査”. 九州大学大学院芸術工学府芸術工学専攻 修士論文, 2022.

[27] Ji-Hoon Kim, Jaehun Kim, and Joon Son Chung. Let there be sound: Reconstructing high quality speech from silent videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, pp. 2759–2767, 2024.

[28] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4879–4883. IEEE, 2018.

[29] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, Vol. 33, pp. 17022–17033, 2020.

[30] Yoshinori Sagisaka, Kazuya Takeda, M Abel, Shigeru Katagiri, Tetsuo Umeda, and Hisao Kuwabara. A large-scale japanese speech database. In *ICSLP*, pp. 1089–1092, 1990.

[31] T Okamoto, Y Shiga, and H Kawai. Hi-fi-captain: High-fidelity and high-capacity conversational speech synthesis corpus developed by nict, 2023.

[32] Shinnosuke Takamichi, Kentaro Mitsui, Yuki Saito, Tomoki Koriyama, Naoko Tanji, and Hiroshi Saruwatari. Jvs corpus: free japanese multi-speaker voice corpus. *arXiv preprint arXiv:1908.06248*, 2019.

[33] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE international conference on computer vision*, pp. 1021–1030, 2017.

[34] Yukiya Hono, Kentaro Mitsui, and Kei Sawada. rinna/japanese-hubert-base.

[35] Kei Sawada, Tianyu Zhao, Makoto Shing, Kentaro Mitsui, Akio Kaga, Yukiya Hono, Toshiaki Wakatsuki, and Koh Mitsuda. Release of pre-trained models for the Japanese language. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 13898–13905, 5 2024. Available at: https://arxiv.org/abs/2404.01657.

[36] Ankita Pasad, Bowen Shi, and Karen Livescu. Comparative layer-wise analysis of self-supervised speech models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[38] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR, 2023.

[39] Simon King, Robert AJ Clark, Catherine Mayo, and Vasilis Karaiskos. The blizzard challenge 2008. 2008.

[40] Ambika Kirkland, Shivam Mehta, Harm Lameris, Gustav Eje Henter, Eva Székely, and Joakim Gustafson. Stuck in the mos pit: A critical analysis of mos test methodology in tts evaluation. In *12th Speech Synthesis Workshop (SSW) 2023*, 2023.

[41] Mirjam Wester, Cassia Valentini-Botinhao, and Gustav Eje Henter. Are we using enough listeners? no! an empirically-supported critique of interspeech 2014 tts evaluations. In *Interspeech 2015*, pp. 3476–3480. International Speech Communication Association, 2015.