日本語 Full-duplex 音声対話システムの試作

大橋厚元　飯塚慎也　姜菁菁　東中竜一郎
名古屋大学大学院 情報学研究科
{ohashi.atsumoto.c0, iizuka.shinya.a8, jiang.jingjing.k6}@s.mail.nagoya-u.ac.jp
higashinaka@i.nagoya-u.ac.jp

概要

人間同士の対話における発話のオーバーラップや相槌など，同時双方向的な特徴をモデル化できる full-duplex 音声対話システムは，近年注目を集めている．しかし日本語においては，full-duplex 音声対話システムはほとんど見られず，full-duplex 音声対話システムの開発に関する知見は不足している．本研究では，英語における主要な full-duplex 音声対話システムである Moshi をベースとすることで，日本語で利用可能な最初の full-duplex 音声対話システムを試作し，公開する．[1)]

1　はじめに

人間同士の自然な音声対話の実現に向け，full-duplex 音声対話システムが注目されている [1, 2, 3]．対話における full-duplex とは，発話のオーバーラップや相槌などの同時双方向的な特徴を示す．互いに相手の発話終了を待ってから応答する従来の対話システム [4, 5] の課題を解決する上で，full-duplex 音声対話システムの研究は必要不可欠である．

Moshi [6] は，代表的な full-duplex 音声対話システムであり，自身とユーザ両方の音声系列を並列にモデル化することで，full-duplex な対話を実現する．その他にも，主に英語において，full-duplex 音声対話システムの研究が増加している [7, 8, 9]．一方，日本語で利用可能な full-duplex 音声対話システムは未だ公開されておらず，英語と比較して，full-duplex 音声対話システムに関する知見は不足している．

本研究の目的は，日本語初の full-duplex 音声対話システムのベースラインを提供することである．本研究では，日本語音声対話データを用いた事前学習およびファインチューニングによって，英語の full-duplex 音声対話システムである Moshi [6] を日本語化する．事前学習では，約 6 万時間のモノラル音声対話を含む J-CHAT コーパス [10] を用いることで，日本語音声対話の基礎能力を獲得させる．そしてファインチューニングでは，2 話者の音声が別々のチャネルで収録された高品質なステレオ音声対話データ 344 時間を用いることで，日本語での完全な full-dupelx 音声対話をモデル化する．人間評価実験を通して，日本語化された Moshi の音声対話生成能力を検証する．

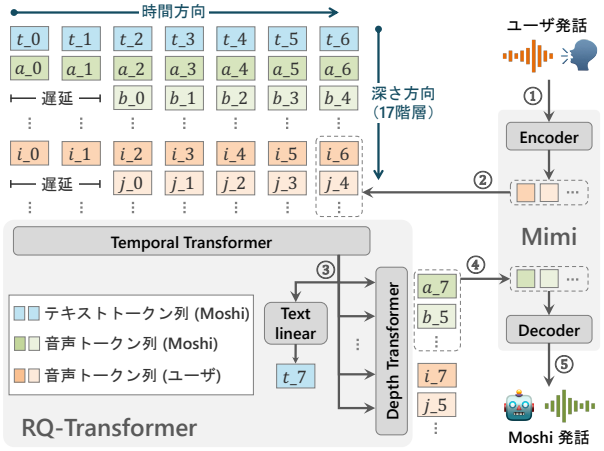

図 1　Moshi のモデル構造．音声波形を離散的な音声トークンにエンコードするニューラル音声コーデック Mimi と，テキストトークンおよび音声トークンの系列を自己回帰的にモデル化する RQ-Transformer から構成される．

2　Moshi

本節では，ニューラル音声コーデック Mimi と，大規模音声言語モデル RQ-Transformer から構成される Moshi [6] のモデル構造（図 1）を説明する．

2.1　ニューラル音声コーデック Mimi

Mimi は，SEANet [11] オートエンコーダと残差ベクトル量子化器 [12] から構成されるニューラル音声コーデック [13] である．エンコーダは，24kHz の音声波形データを 12.5Hz のフレームレートで音声

1）学習済みモデルおよび生成音声のサンプルは https://nu-dialogue.github.io/j-moshi で公開している．

トークンに離散化する．各フレームは 8 階層のトークンで構成され，第 1 階層のトークンは音声の意味情報を，第 2〜8 階層は音響情報を保持するように学習される．第 1 階層は意味トークン，第 2〜8 階層は音響トークンと呼ばれる．

図 1 に示すように，Mimi はユーザの入力音声をエンコードしてユーザの音声トークンを出力し RQ-Transformer に流す．同時に，RQ-Transformer から生成された Moshi の音声トークンをデコードすることで，Moshi の出力音声発話を生成する．

2.2 音声言語モデル RQ-Transformer

RQ-Transformer は，7B の大規模言語モデル（LLM）をベースとした Temporal Transformer と，より小規模な Depth Transformer から構成される．Temporal Transformer は，時間方向に沿ったトークン列を 12.5Hz のレートでモデル化する．トークン列には，Moshi のテキスト発話トークン列（1 階層），Moshi の音声トークン列（8 階層），そしてユーザの音声トークン列（8 階層）の計 17 階層が含まれる．各時間ステップ s において，直前のステップ $s-1$ までの $(s-1)\times 17$ トークンから，埋め込みベクトル z_s を出力する．続けて，Text Linear 層によって，s におけるテキストトークンを z_s からサンプルする．

Depth Transformer は，s における音声トークンを深さ方向に沿ってモデル化する．z_s を入力として，Moshi の音声トークン 8 個とユーザの音声トークン 8 個を自己回帰的にサンプルする．以上のように，Moshi は，LLM の高い言語能力を活用することで，自然な音声対話生成を実現する．なお，生成される音声の品質を安定化させるため，音響トークンには 1 時間ステップ分の遅延を設けている．

一般に，単位時間におけるテキスト列の長さは，音声トークン列の長さに比べ短い．そこで，テキストトークンと音声トークンのアライメントを取るため，学習データ作成時，Whisper [14] によって作成できるトークン単位の書き起こし（各テキストトークンがどの時間ステップに対応するかの情報）を活用し，テキストトークンが割り当てられないタイムステップには PAD トークンを埋める処理が行われる．

Moshi の学習は，700 万時間のモノラル音声対話による事前学習，2 千時間のステレオ音声対話によるファインチューニング，そして音声合成器（TTS）を用いて生成された 2 万時間のステレオ音声対話によるインストラクションチューニングからなる．以降は，この学習済みモデルを英語版 Moshi と呼ぶ．

表 1 Moshi の日本語化に使用された対話データの一覧

コーパス名	対話数	時間
事前学習用データ		**68,892**
J-CHAT [10]	4,937,497	68,892
ファインチューニング用データ		**344**
日本語 Callhome [16]	120	16
CSJ [17]（対話音声のみ）	58	12
旅行代理店対話コーパス [18]	330	115
雑談対話コーパス	500	148
相談対話コーパス	100	53
Multi-stream TTS による合成音声データ		**602**
日本語 PersonaChat [19]	4,983	94
日本語 EmpatheticDialogues [19]	20,000	102
日本語日常対話コーパス [20]	5,246	44
RealPersonaChat [21]	13,510	362

3 日本語での追加学習

本節では，英語版 Moshi[2] を日本語化する上で実施した，テキスト語彙の日本語化，そして事前学習とファインチューニングからなる 2 段階の学習ステップを説明する．また，TTS によるデータ拡張についても説明する．なお，Mimi は追加学習なしでも，日本語音声の再合成がある程度可能であったことから，Mimi のパラメータは凍結し，RQ-Transformer のパラメータのみを学習した（Mimi の日本語性能については 4.2 および A.4 節に示す）．

3.1 テキスト語彙の日本語化

英語版 Moshi のテキストトークナイザは英語データから学習された SentencePiece [15] であり 32,000 語彙を持つが，日本語語彙は含まれないため，日本語テキストのトークナイズには非効率である．そこで本研究では，日本語 GPT-2[3] の SentencePiece モデルを日本語トークナイザとして採用した．また，トークナイザの交換に伴い，テキスト語彙に紐づく RQ-Tranformer の一部重みを初期化した．具体的には，Temporal Tranformer と Depth Transformer におけるテキストトークン埋め込みテーブル，および，Text Linear のパラメータをランダム初期化した．

3.2 事前学習

事前学習の目的は，大規模な日本語音声対話データによって，日本語音声対話の基盤能力を獲得することである．本研究では，YouTube および Podcast から収集された 6.9 万時間の日本語音声対話を含む

2） https://huggingface.co/kyutai/moshika-pytorch-bf16
3） https://huggingface.co/rinna/japanese-gpt2-medium

J-CHAT コーパス [10] を採用した．

J-CHAT の前処理　J-CHAT の音声データは，全ての話者の音声が同じチャネルに収録されたモノラル音声であるため，2 チャネル（すなわち Moshi 自身とユーザ）の音声対話モデルに使用できない．そこで，英語版 Moshi と同様に，各音声に話者分離[4]を施し，ランダムに選ばれた 1 話者を Moshi のチャネル，それ以外をユーザのチャネルとしたステレオ音声対話を作成した．次に，各チャネルの書き起こしを音声認識器（ASR）[5]によって作成した．さらに WhisperX [22] によるトークン単位の書き起こしを基に，PAD トークンを用いたテキストと音声のアライメントを作成した．最終的に得られたデータには，合計で 30 億のテキストトークンが含まれ，その内の PAD トークンの割合は，約 88%であった．

J-CHAT での学習　前処理された J-CHAT のうち train セット（約 6 万時間）を 1 エポック学習した．深層学習ライブラリ DeepSpeed で実装された ZeRO-3 データ並列 [23] を採用し，128 基の NVIDIA V100 32GB GPU で学習を実施した．混合精度（float16）および，各 Transformer 層への activation checkpointing を使用した．各入力サンプルの最大長は 2.7 分（時間方向で 2,048 トークン）とし，合計バッチサイズは 512 サンプルとした．AdamW [24] を使用し，Llama 2 7B [25] に倣い，$\beta_1 = 0.9$，$\beta_2 = 0.95$，$\epsilon = 1e-5$，そして重み減衰を 0.1 に設定した．学習率は $3e-5$ とし，500 ステップの線形 warmup を使用した．英語版 Moshi と同様に，損失計算時の重み付けでは，PAD トークンの損失は 50%軽減し，意味トークンと音響トークンの損失比率は 100 : 1 とした．合計の最適化ステップ数は 8,880 となり，36 時間を要した．付録の図 2 に学習時の損失曲線を示す．

3.3　ファインチューニング

事前学習で用いられた学習データは，モノラル音声を強制的に 2 チャネルに分割したものであるため，発話のオーバーラップや相槌など，自然なターンテイキングが含まれていない．実際の full-duplex 音声対話をモデル化するため，チャネルごとに各話者の音声が収録されたステレオ音声対話データを用いたファインチューニングを行った．比較的サイズの大きいステレオ音声対話コーパスとして，日本語 Callhome [16]，CSJ [17]，そして旅行代理店対話コーパス [18] の 3 種類を採用した．さらに，研究室内で作成した Zoom 通話による雑談対話コーパスと相談対話コーパスを追加し，合計で 344 時間の音声対話データを用意した．表 1 にデータの内訳を示す．各コーパスの説明は，A.1 節に示す．

合計 344 時間の上記データを，3.2 節と同様の手法でトークナイズし，train/valid/test セットを 94 : 3 : 3 の比率で分割した．そして train セットに含まれる 320 時間分のデータを 4 基の NVIDIA V100 32GB GPU で 3 エポック学習した．基本的なハイパーパラメータは事前学習時と同じであるが，合計バッチサイズは 16 サンプルとし，Temporal Transformer と Depth Transformer の学習率は $2e-6$ と $4e-6$ とした．合計の最適化ステップ数は 1,416 であった．以降では，ここで得られたモデルを J-Moshi と呼ぶ．

3.4　Multi-stream TTS によるデータ拡張

英語版 Moshi は，ファインチューニング後，テキスト対話から multi-stream TTS [6] によって合成された 2 万時間分のステレオ音声対話を用いてさらに学習された．そこで本研究でも，multi-stream TTS によってテキスト対話から音声対話を合成し，Moshi のファインチューニング用データに含める．これにより，学習データに含まれる対話の多様性を高め，より汎用的な対話能力を獲得することが期待される．Défossez ら [6] の実装に倣い，Moshi の意味トークンの遅延を 25 に，音響トークンの遅延を 27 に設定した後，J-CHAT による事前学習，および 344 時間のステレオ音声対話データによるファインチューニングを行うことで multi-stream TTS を実装した．学習設定は全て 3.2 節，および 3.3 節と同様である．以降では，合成音声の元となるテキスト対話データ，および音声合成手順について説明する．

テキスト対話データの準備　Multi-stream TTS によって音声化するテキスト対話データとして，本研究では，既存のテキスト対話コーパス，すなわち日本語 PersonaChat [19]，日本語 EmpatheticDialogues [19]，日本語日常対話コーパス [20]，そして RealPersonaChat [21] の 4 つを使用した．これらコーパスは，テキストチャット形式で収集されたため，書き言葉の表現を多く含み，話し言葉である対話音声を合成する目的には適さない．そこで，先行研究 [5] に倣い，話し言葉特有の表現を含むように，LLM [6]を用いてテキスト対話の書き換

4）https://huggingface.co/pyannote
5）https://huggingface.co/reazon-research/reazonspeech-espnet-v2
6）https://huggingface.co/google/gemma-2-27b-it

えを実施した．結果的に上記 4 つのコーパスに含まれる全 43,739 対話を書き換えた．

ステレオ音声対話の生成　Multi-stream TTS を用い，前段で得られた各テキスト対話に対して異なるシード値で 10 個の音声サンプルを生成した（生成における詳細な設定は付録の A.2 節に示す）．そして，この 10 サンプルの中から，ASR 結果と元の対話テキストとの単語誤り率（WER）が最も低いサンプルを，その対話の最終的な音声サンプルとして採用した．結果的に 602 時間のステレオ音声対話が合成された．データ全体の WER は 24.6%であった．表 1 に合成音声データの内訳を示す．

合成された音声対話データを追加して得られた，合計 946 時間の音声データを用いて，J-CHAT で事前学習済みの Moshi をファインチューニングした．学習設定は 3.3 節と同様であり，合計の最適化ステップは 2,401 であった．以降では，この拡張データによって学習されたモデルを J-Moshi-ext と呼ぶ．

4　評価実験

Full-duplex 音声対話モデルの評価に一般的に用いられる対話継続タスク [1, 7, 6] を採用し，J-Moshi および J-Moshi-ext の音声対話生成の性能を主観評価により検証した．対話継続タスクとは，数秒の対話音声をプロンプトとし，その続きを生成するタスクである．本研究では，ファインチューニング用データ（3.3 節を参照）の test セットに含まれる各対話音声を 30 秒間隔で分割し，結果として得られた 709 個の音声サンプルそれぞれについて，最初の 10 秒をプロンプトとし，続く 20 秒をモデルに生成させた．

各モデルが出力した 709 個の対話音声から，それぞれ 50 個の音声サンプルをランダムに抽出した．クラウドソーシング[7)] を介して合計 125 人の評価者を応募し，1 人あたり 10 個の音声サンプルを評価した．先行研究 [1, 10] に倣い，評価軸 2 つ，すなわち，自然性（人間のような自然な対話に聞こえるか）と意味性（音声の意味がわかるかどうか）をそれぞれ 5 段階で評価した．なお，付録の A.4 節に，人間評価に先立って実施した自動評価実験について示す．

4.1　ベースライン

Moshi の比較対象として，最も標準的な full-duplex 音声対話モデルである dGSLM [1] を採用した．dGSLM の学習では，J-Moshi と同様に，J-CHAT による事前学習と 344 時間のステレオ音声対話データによるファインチューニングを実施した．dGSLM の実装，および，学習設定の詳細は A.3 節に示す．

日本語音声対話における Mimi の性能を評価するため，実際の 20 秒の音声を Mimi によって単に再合成したもの（Re-synthesis）と，本物の 20 秒の音声（Ground-truth）をベースラインに含めた．

表 2　生成された音声対話の 5 段階評価スコアと 95%信頼区間．τ は生成時の温度パラメータを示す．

Model	τ	自然性	意味性
dGSLM		2.44±0.12	1.76±0.09
J-Moshi	0.8	**2.67±0.13**	2.19±0.12
J-Moshi-ext	0.8	2.66±0.13	**2.30±0.13**
Re-synthesis		3.90±0.12	3.92±0.13
Ground-truth		4.46±0.09	4.45±0.10

4.2　人間評価結果

表 2 に結果を示す．自然性においては，dGSLM よりも J-Moshi および J-Moshi-ext が高性能であった．また意味性においても dGSLM の性能を大幅に上回っていた．特に J-Moshi と比較して，J-Moshi-ext はさらに意味性が改善しており，これは，multi-stream TTS が言語能力改善に寄与したことを示している．一方で，Re-synthesis と比較すると，自然性と意味性の両方が 1 ポイント以上悪化しており，RQ-Transformer における改善の余地は大きいことがわかる．また Re-synthesis のスコアは，Ground-truth に対して約 0.5 ポイント劣化しており，今後は Mimi の日本語化も重要となる．

5　結論と今後の展望

本研究では，日本語の full-duplex 音声対話システムとして，英語版 Moshi を日本語化した J-Moshi を構築・公開した．6.9 万時間の J-CHAT コーパスによる事前学習と，344 時間のステレオ音声対話データによるファインチューニングを行い，さらに multi-stream TTS を用いた 602 時間の合成データによる性能改善を試みた．実験では，J-Moshi が生成した対話音声の自然性および意味性を評価した．なお，本研究では最初のステップとして対話継続タスクのみを実施したが，実際のユーザからの音声トークンを入力することで，J-Moshi とのリアルタイムの対話は容易に実現できる．そこで今後は，人間とのインタラクティブな評価によって，J-Moshi の対話システムとしての性能を検証したい．

7）https://crowdworks.jp/

謝辞

本研究は，JST ムーンショット型研究開発事業，JPMJMS2011 の支援を受けた．雑談対話コーパス，および，相談対話コーパスは，株式会社アイシンとの共同研究において構築した．また，本研究では，名古屋大学のスーパーコンピュータ「不老」を利用した．

参考文献

[1] Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoît Sagot, Abdelrahman Mohamed, and Emmanuel Dupoux. Generative Spoken Dialogue Language Modeling. **Transactions of the Association for Computational Linguistics**, pp. 250–266, 2023.

[2] Peng Wang, Songshuo Lu, Yaohua Tang, Sijie Yan, Wei Xia, and Yuanjun Xiong. A Full-duplex Speech Dialogue Scheme Based On Large Language Model. In **The Thirty-eighth Annual Conference on Neural Information Processing Systems**, 2024.

[3] Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. Language Model Can Listen While Speaking. **arXiv preprint arXiv:2408.02622**, 2024.

[4] Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities. In **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 15757–15773, 2023.

[5] Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. Llama-omni: Seamless speech interaction with large language models. **arXiv preprint arXiv:2409.06666**, 2024.

[6] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. **arXiv preprint arXiv:2410.00037**, 2024.

[7] Bandhav Veluri, Benjamin N Peloquin, Bokai Yu, Hongyu Gong, and Shyamnath Gollakota. Beyond Turn-Based Interfaces: Synchronous LLMs as Full-Duplex Dialogue Agents. In **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 21390–21402, 2024.

[8] Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, and Chaohong Tan. Omniflatten: An end-to-end gpt model for seamless voice conversation. **arXiv preprint arXiv:2410.17799**, 2024.

[9] Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. SALMONN-omni: A Codec-free LLM for Full-duplex Speech Understanding and Generation. **arXiv preprint arXiv:2411.18138**, 2024.

[10] Wataru Nakata, Kentaro Seki, Hitomi Yanaka, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari. J-CHAT: Japanese Large-scale Spoken Dialogue Corpus for Spoken Dialogue Language Modeling. **arXiv preprint arXiv:2407.15828**, 2024.

[11] Marco Tagliasacchi, Yunpeng Li, Karolis Misiunas, and Dominik Roblek. SEANet: A multi-modal speech enhancement network. **arXiv preprint arXiv:2009.02095**, 2020.

[12] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, Vol. 30, pp. 495–507, 2021.

[13] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High Fidelity Neural Audio Compression. **Transactions on Machine Learning Research**, 2023.

[14] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision. In **Proceedings of the 40th International Conference on Machine Learning**, pp. 28492–28518, 2023.

[15] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 66–71, 2018.

[16] Alexandra Canavan and George Zipperlen. CALLHOME Japanese Speech LDC96S37. Philadelphia: Linguistic Data Consortium, 1996.

[17] Kikuo Maekawa. Corpus of Spontaneous Japanese: Its design and evaluation. In **Proceedings of ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition**, 2003.

[18] Michimasa Inaba, Yuya Chiba, Zhiyang Qi, Ryuichiro Higashinaka, Kazunori Komatani, Yusuke Miyao, and Takayuki Nagai. Travel agency task dialogue corpus: A multimodal dataset with age-diverse speakers. **ACM Transactions on Asian and Low-Resource Language Information Processing**, Vol. 23, No. 9, pp. 1–23, 2024.

[19] Hiroaki Sugiyama, Masahiro Mizukami, Tsunehiro Arimoto, Hiromi Narimatsu, Yuya Chiba, Hideharu Nakajima, and Toyomi Meguro. Empirical Analysis of Training Strategies of Transformer-Based Japanese Chit-Chat Systems. In **Proceedings of 2022 IEEE Spoken Language Technology Workshop**, pp. 685–691, 2023.

[20] 赤間怜奈, 磯部順子, 鈴木潤, 乾健太郎. 日本語日常対話コーパスの構築. 言語処理学会 第 29 回年次大会 発表論文集, pp. 108–113, 2023.

[21] Sanae Yamashita, Koji Inoue, Ao Guo, Shota Mochizuki, Tatsuya Kawahara, and Ryuichiro Higashinaka. RealPersonaChat: A realistic persona chat corpus with interlocutors' own personalities. In **Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation**, pp. 852–861, 2023.

[22] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. WhisperX: Time-Accurate Speech Transcription of Long-Form Audio. **arXiv preprint arXiv:2303.00747**, 2023.

[23] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: memory optimizations toward training trillion parameter models. In **Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis**, 2020.

[24] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In **International Conference on Learning Representations**, 2019.

[25] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. **arXiv preprint arXiv:2307.09288**, 2023.

[26] 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.

[27] Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. Speech Resynthesis from Discrete Disentangled Self-Supervised Representations. **arXiv preprint arXiv:2104.00355**, 2021.

A 付録

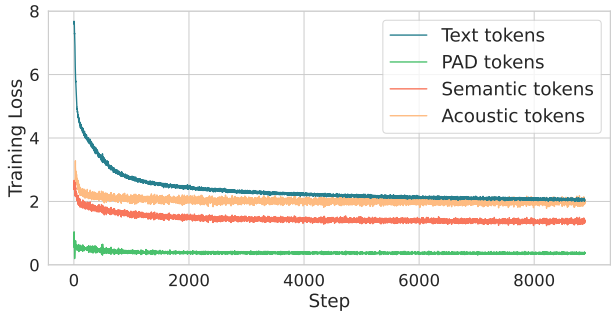

図 2 J-CHAT での事前学習における J-Moshi の損失曲線

A.1 ステレオ音声対話コーパス

日本語 Callhome [16] 電話での雑談対話が収録されたコーパス．書き起こしが不足している一部音声を除き，残りの 16 時間を使用した．

CSJ [17] 日本語での話し言葉の音声が合計 660 時間収録されたコーパス．2 話者による音声対話が収録された 12 時間のみを使用した．

旅行代理店対話コーパス [18] 旅行代理店のオペレータ役と顧客役による相談対話を Zoom を介して収録されたコーパス．合計 115 時間の対話音声を含む．

雑談対話コーパス 研究室内で作成された雑談対話コーパス．各対話は Zoom を介して収録された．32 人の対話者による合計 148 時間の対話音声を含む．

相談対話コーパス 雑談対話コーパスと同様に，研究室内で作成された相談対話コーパス．32 人の対話者による合計 53 時間の対話音声を含む．

A.2 Multi-stream TTS による音声合成

テキスト対話データから合成されるステレオ音声のうち，Moshi に相当する第 1 チャネルの声には一貫性を持たせるべきである．そこで，ファインチューニングデータ中のある話者の発話「よろしくお願いします」からエンコードされた音声トークン列を，multi-stream TTS の第 1 チャネルの prefix として入力することで，生成される第 1 チャネルの声を制御した．音声生成時，テキストトークンと音声トークンをサンプルする際の温度パラメータは，それぞれ 0.55 と 0.60 とし，top-p はいずれも 1.0 とした．

A.3 dGSLM の詳細

dGSLM は，音声波形を音声トークンにエンコードする HuBERT [26] ベースの speech-to-unit（s2u），2 チャネルの音声トークン系列を並列して自己回帰的にモデル化する unit language model（uLM），そして音声トークンを音声波形に復元する HiFi-GAN [27] ベースの unit-to-speech（u2s）で構成される．dGSLM は，入力音声のエンコードに非自己回帰型の HuBERT を用いるため，実際に対話を行うことはできない proof-of-concept なモデルではあるが，対話継続タスクによる評価は可能である．

dGSLM の uLM については，Nguyen ら [1] の実装，および，学習設定を採用した．uLM の学習においては，J-Moshi と同様の手順，すなわち J-CHAT による事前学習と 344 時間のステレオ音声対話データによるファインチューニングを実施した．s2u の基盤モデルには，日本語 HuBERT[8] を採用した．s2u および u2s の実装には音声処理ツールキット SpeechBrain[9] を使用し，雑談対話コーパスと相談対話コーパスに含まれる 201 時間のステレオ音声対話データをチャネルごとに別の音声サンプルとして分割した合計 402 時間のデータを学習に利用した．

表 3 生成された音声対話の自動評価結果．τ は生成時の温度パラメータを示す．

Model	τ	PPL ↓	IPU	Pause	Gap	Overlap
dGSLM		305.63	57.74	4.44	3.20	6.19
J-Moshi	0.8	**197.69**	53.24	6.26	4.48	4.98
	0.9	268.53	61.33	3.80	3.43	9.12
	1.0	345.43	70.64	2.38	2.25	15.70
J-Moshi-ext	0.8	**209.41**	50.93	7.01	4.55	4.15
	0.9	282.71	59.95	3.94	3.59	8.12
	1.0	370.83	68.80	2.53	2.39	14.11
Re-synthesis		74.28	60.24	3.27	3.97	8.29
Ground-truth		56.81	59.70	3.52	4.03	8.05

A.4 自動評価実験

自動評価実験では，各モデルが生成した対話音声の ASR 結果の流暢性を，言語モデル[10] の perplexity（PPL）によって計測した．また，Nguyen ら [1] が用いたターンテイキングに関する 4 つの統計量，すなわち，最低 0.2 秒の無音で区切られた発話音声の時間（Inter-Pausal Units; IPU），同一話者からの IPU 間の無音時間（Pause），異なる話者の IPU 間の無音時間（Gap），そして，異なる話者の IPU が重なる時間（Overlap）も計測した．なお，IPU，Pause，Gap，Overlap はいずれも 1 分間における累積時間である．

結果を表 3 に示す．J-Moshi および J-Moshi-ext の PPL から，英語版 Moshi [6] と同様に，トークンサンプルにおける温度パラメータ τ が小さい方がより流暢な発話を生成できることが判明した．$\tau = 0.8$ の設定では，フルスクラッチから日本語データで dGSLM よりも PPL が 100 ポイントほど改善している．このことから，発話流暢性の観点では，英語版 Moshi の日本語化に一定の効果があったと言える．一方で，Re-synthesis や Ground-truth と比較すると，PPL は大きく悪化しており，IPU と Overlap は短くなっている．対話の流暢さとターンテイキング能力を改善することは今後の課題である．

J-Moshi-ext の PPL は，J-Moshi と比較して若干悪化していた．理由としては，multi-stream TTS が発音できない文字がテキスト対話コーパスに多く含まれており，流暢な音声対話を合成できなかったことが考えられる．日本語では，漢字など文字の種類が多いことに加え，文脈によって発音が変わる漢字（“々” など）などを適切に合成できていない事例が散見された．本研究で用いた学習データだけでは，多様なテキストを音声合成できるだけの十分な事例を網羅できなかった可能性がある．

Ground-truth と比較した場合の，Re-synthesis の PPL の劣化，およびターンテイキングの統計量の変化は，最小限であることがわかる．このことは，Moshi の日本語化の初期段階においては，Mimi は日本語音声対話にそのまま適用可能であることを示唆している．

8） https://huggingface.co/rinna/japanese-hubert-base
9） https://github.com/speechbrain/speechbrain
10） https://huggingface.co/llm-jp/llm-jp-3-3.7b