Université de Montréal

Towards a Unified Model for Speech and Language Processing

par

Artem Ploujnikov

Département d'informatique et de recherche operationelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Intelligence artificielle

August 31, 2023

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Towards a Unified Model for Speech and Language Processing

présenté par

Artem Ploujnikov

a été évalué par un jury composé des personnes suivantes :

Simon Lacoste-Julien

(président-rapporteur)

Mirco Ravanelli

(directeur de recherche)

Bang Liu

(membre du jury)

Résumé

Ce travail de recherche explore les méthodes d'apprentissage profond de la parole et du langage, y inclus la reconnaissance et la synthèse de la parole, la conversion des graphèmes en phonèmes et vice-versa, les modèles génératifs, visant de reformuler des tâches spécifiques dans un problème plus général de trouver une représentation universelle d'information contenue dans chaque modalité et de transférer un signal d'une modalité à une autre en se servant de telles représentations universelles et à générer des représentations dans plusieurs modalités. Il est compris de deux projets de recherche: 1) SoundChoice, un modèle graphème-phonème tenant compte du contexte au niveau de la phrase qui réalise de bonnes performances et des améliorations remarquables comparativement à un modèle de base et 2) MAdmixture, une nouvelle approche pour apprendre des représentations multimodales dans un espace latent commun.

Mot-clés: parole — apprentisage de représentations — reconaissance vocale — synthèse vocale — graphème-phonème – G2P

Abstract

The present work explores the use of deep learning methods applied to a variety of areas in speech and language processing including speech recognition, grapheme-to-phoneme conversion, speech synthesis, generative models for speech and others to build toward a unified approach that reframes these individual tasks into a more general problem of finding a universal representation of information encoded in different modalities and being able to seamlessly transfer a signal from one modality to another by converting it to this universal representations and to generate samples in multiple modalities. It consists of two main research projects: 1) SoundChocice, a context-aware sentence level Grapheme-to-Phoneme model achieving solid performance on the task and a significant improvement on phoneme disambiguation over baseline models and 2) MAdmixture, a novel approach to learning a variety of speech representations in a common latent space.

Keywords: speech — representation learning — deep learning — speech recognition — ASR — text-to-speech — TTS — grapheme-to-phoneme – G2P

Contents

List of tables

List of figures

List of abbreviations

ASR	Automatic Speech Recognition
CRDNN	Convolutional, Recurrent and Deep Neural Network
CTC	Connectionist Temporal Classification
G2P	Grapheme-to-Phoneme
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
MSE	Mean Squared Error
NLL	Negative Log Likelihood
NLU	Natural Language Understanding
PDF	Probability Density Function
PER	Phoneme Error Rate

RNN	Recurrent Neural Network
STFT	Short-Term Fourier Transform
TER	Token Error Rate
TTS	Text-to-Speech

Acknowledgments

I would like to thank Prof. Mirco Ravanelli and all present and past members of the Mila-Concordia Conversational AI Lab for creating an extremely supportive open learning and idea-sharing environment, as well as Mila, OVH and the Digital Research Alliance of Canada for providing computational resources. I would also like to thank the entire Mila and UdeM community for supporting me throughout my journey as an M.Sc. student.

Introduction

Speech processing is one of the most widely-used and well-studied applications of machine learning and artificial intelligence with a variety of uses in industry ranging from digital assistants to automatic transcripts for call centers to subtitles to automatic translation. Most of us interact with speech processing systems daily in some capacity. Typical tasks performed with speech processing include speech recognition (converting speech to written text), speech synthesis (converting written text to speech), natural language understanding (converting text and/or speech to actionable commands to be executed by an information system), speaker separation (disentangling the overlapping speech of several speakers), speaker identification and end-to-end translation.

Most early work on the subject treated each task completely in isolation, with a dedicated set of methods used for each task relying on tailor-made feature-engineered representations. The earliest methods [**5**] of speech recognition relied on locating formants within the audio spectrum. Subsequent incremental improvements to the technologies involved the use of more sophisticated statistical modeling techniques, such Dynamic Time Warping (DTW) [**14**], which was eventually superseded by an approach based on Hidden Markov Models (HMM) [**15**] representing speech as piecewise stationary signals. While these approaches constituted a notable improvement, they still imposed strong assumptions on the signal and were not very usable for practical applications in user interfaces. While consumer products based on such technologies were usually successful at recognizing specific user commands, they often required model fine-tuning for a specific features, and they also struggled with foreign-accented speech or non-typical voices. Commercial products built in the 1990s were unable to recognize fluent speech, requiring the speaker to pause between words and not allowing for any background noise. At the time popular consumer-grade voice assistants were introduced, speech recognition technology still relied primarily on classical methods, resulting in infamously inaccurate command recognition. In recent years, with the advent of modern deep learning-based methods [**12, 20**] and off-device speech recognition allowing for more compute resources to be used for the task, consumer-grade speech recognition for simple commands is finally at a level where it can be used as a primary interface for smart home applications, achieving reasonable accuracy without a significant effort on the part of the user. Other key components of the processing pipeline, such as natural language understanding [**23**, **44**], are still active research areas, and in recent years, advances in large language models and their use as foundation models for other tasks have contributed to progress in NLU [**7**].

Speech synthesis follows a distinct, independent, yet somewhat similar development trajectory. Intrinsically, the task is more mechanical and predetermined than speech recognition and therefore, has been possible to implement in rudimentary forms prior to recent advances in artificial intelligence, and even before the invention of the computer. The earliest attempts at recreating human speech using machines predate modern digital computing technology with Christian Gottlieb Kratzenstein [**24**] building a physical model of the human vocal tract in 1779, producing distinct vowel sounds. Subsequently, a mechanical machine producing both consonants and vowels was proposed by Wolfgang von Kempelen [**31**] in 1791 and built by Charles Wheatstone in 1837. The first electronic vocoder was built by Bell Labs in the 1930s, utilizing representations of speech consisting of fundamental tones and resonances, which parallels some of the methods that are still in use. The first computer-based speech synthesis work started in the 1950s, culminating in the first complete text-to-speech system for English was developed by Noriko Umeda in 1968, still relying on an anatomical model of human speech organs - but represented algorithmically rather than with a physical device. Further improvements were achieved with approaches using statistical methods. Linear Predictive Coding [**25**, **29**] was used in early speech synthesis chips The deep learning [**19**] revolution has made inroads into the speech synthesis space as well with deep neural networks gradually replacing feature engineering combined with classical statistical methods. The original seminal work in using deep learning [**19**] methods for speech synthesis was WaveNet [**32**], a deep convolutional network relying on dilated convolutions to create a sufficient receptive field for long sequences and a gated unit inspired by PixelCNN [**33**]. In deep-learning text-to-speech systems, WaveNet is commonly used as a vocoder, i.e. the component that converts a compact representation of speech, such as a MEL spectrogram, to a raw waverform. A variety of deep learning approaches have been attempted for deep learning, including the Recurrent Neural Network-based Tacotron [**8**,**19**], which leverages an attention mechanism, the compute efficiency-driven convolutional DeepVoice [**1**, **2**, **27**] family, also leveraging an encoder-decoder architecture with attention but requiring significantly less training time and, more recently, the TransformerTTS [**20**] model, leveraging the recurrence-free, parallelizable Transformer [**41**] architecture, which has demonstrated solid performance exceeding that of earlier models on a variety of sequential tasks. More recently, denoising diffusion probabilistic models [**12**] have achieved state-of-the-art performance on a number of generative tasks. While most initial applications of the technique focussed primarily on photographic image generation, the DiffTTS [**13**] model has successfully applied the technique to waveform generation.

Grapheme-to-Phoneme models are used to determine the pronunciations of words or entire sentences symbolically, converting from a representation based on a written alphabet to one derived from a phonetic transcription alphabet, such as ARPABET or IPA. Such models can be used either in isolation or in the processing pipeline of a text-to-speech system, such as the G2P-based Tacotron2 [**8**] varieties, and they can, in theory, be particularly helpful for dealing with languages with a highly irregular non-phonetic spelling, such as English and French. Early approaches, such as the one introduced by Warmus and Bole in 1973 [**36**], relied on deterministic algorithms. The task is easily amenable to deep learning models designed for sequence processing, such as RNNs and Transformers, and neural approaches to G2P were introduced shortly after the usage of these architectures became widespread for other tasks - using LSTM RNNs [**13**], convolutional networks [**21**] and, more recently, Transformers [**22**]. Recent research directions include massively multilingual G2P systems, such as ByT5 [**40**] and sentence-level ones, such as T5G2P [**30**]. The present work explores the use of mixed representations combined with a variety of techniques used in speech recognition for sentence-level G2P with semantic disambiguation.

Originally, most machine learning models were trained for a specific tasks. Such is the case with most of the early state-of-the-art models driving the deep learning revolution, including image classification models, such as AlexNet [**17**] and VGG [**30**]. That was also the case with most early speech and language processing models, such as the ones listed previously, and numerous current ones as well. The hallmark of human and animal intelligence and learning is integrating inputs from multiple modalities to form a unified picture of the world. Recently, many advances have been made in multimodal AI with deep neural networks combining representations of a single message or piece of information in two or more modalities. For instance, Facebook's data2vec [**3**] combines speech, text and video, while MaMMUT [**18**] proposes a unified model for text and images. Many common speech processing scenarios lend themselves to the multimodal setting with the same signal or message being encoded in different modalities, such as speech, text, phonetic transcriptions, etc that can all be combined in a representation-sharing model.

In many deep learning models, especially ones related to sequential tasks, alignment between inputs and outputs is of utmost importance. Early sequential deep learning models, such as language models [**23**] and neural language translation had difficulty maintaining context and alignment on longer sequences, frequently suffering from vanishing and exploding gradient problems [**26**], particularly when relying exclusively on recurrences to maintain context. Attention mechanisms were gradually introduced to mitigate these deleterious effects, starting with Bahdanau attention [**4**], which relies on adding weighted states from both the encoder and the decoder, with weights being learned separately. Subsequently, Luong et al [**21**] have introduced an alternate multiplicative approach where the output is determined either by a dot product of encoder and decoder states (with optional learned weights) or a concatenation. Transformer [**41**]-based models have subsequently superseded recurrent neural networks for a wide variety of sequence tasks, including speech processing [**6**] [**20**], featuring an easily parallelizable dot-product attention mechanism and a recurrence-free architecture. Additionally, Connectionist Temporal Classification [**11**] is a common alignment-free technique, which is fundamentally based on integrating over the probabilities of all possible alignments - with a branch of the model used to estimate the alignment probability - and a special loss function that efficiently sums up possible alignments using a dynamic programming algorithm [**10**]. Combining CTC with an attention mechanism during decoding has been successfully applied to speech recognition [**16**]. The present work takes advantage of both attention and CTC for training and for sequence decoding - and it explores a constrained alternative to traditional attention for cross-modality alignment.

Fundamentally, speech processing involves a variety of generative tasks, such as synthesis, discriminative tasks operating across different modalities, such as speech recognition and speaker identification and others. The main goal of the present work is to create a unified model combining both generative and discriminative capabilities by leveraging a common latent space for multiple speech representations and a variety of mechanisms for encoding and decoding multiple representations of speech - or, more generally, of given information represented in one or more modalities. While this study focuses mainly on speech, the methods proposed here are rather general and can be extended to various other applications.

References

[1] Sercan Ömer Arik, Mike Chrzanowski, Adam Coates, Greg Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. Deep voice: Real-time neural text-to-speech. *CoRR*, abs/1702.07825, 2017.

[2] Sercan Ömer Arik, Gregory F. Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech. *CoRR*, abs/1705.08947, 2017.

[3] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. *CoRR*, abs/2202.03555, 2022.

[4] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. January 2015. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

[5] KH Davis, R Biddulph, and Stephen Balashek. Automatic recognition of spoken digits. *Journal of the Acoustical Society of America*, 24(6):637–642, 1952.

[6] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888, 2018.

[7] Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. Shortcut learning of large language models in natural language understanding: A survey. *CoRR*, abs/2208.11857, 2022.

[8] Isaac Elias, Heiga Zen, Jonathan Shen, Yu Zhang, Jia Ye, R. J. Skerry-Ryan, and Yonghui Wu. Parallel tacotron 2: A non-autoregressive neural TTS model with differentiable duration modeling. *CoRR*, abs/2103.14574, 2021.

[9] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA, 2006. Association for Computing Machinery.

[10] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In William W. Cohen and Andrew W. Moore, editors, *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376. ACM, 2006.

[11] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1764–1772, Bejing, China, 22–24 Jun 2014. PMLR.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[13] Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. Diff-tts: A denoising diffusion model for text-to-speech, 2021.

[14] B.-H. Juang. On the hidden markov model and dynamic time warping for speech recognition — a unified view. *ATT Bell Laboratories Technical Journal*, 63(7):1213–1243, 1984.

[15] B. H. Juang and L. R. Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[16] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. *CoRR*, abs/1609.06773, 2016.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[18] Weicheng Kuo, AJ Piergiovanni, Dahun Kim, Xiyang Luo, Ben Caine, Wei Li, Abhijit Ogale, Luowei Zhou, Andrew Dai, Zhifeng Chen, Claire Cui, and Anelia Angelova. Mammut: A simple architecture for joint learning for multimodal tasks, 2023.

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[20] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu, and Ming Zhou. Close to human quality TTS with transformer. *CoRR*, abs/1809.08895, 2018.

[21] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[22] Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. Dialoglue: A natural language understanding benchmark for task-oriented dialogue, 2020.

[23] Tomás Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048. ISCA, 2010.

[24] John J. Ohala. Christian gottlieb kratzenstein: Pioneer in speech synthesis. In *International Congress of Phonetic Sciences*, 2011.

[25] D. O'Shaughnessy. Linear predictive coding. *IEEE Potentials*, 7(1):29–32, 1988.

[26] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063, 2012.

[27] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan Ömer Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: 2000-speaker neural text-to-speech. *CoRR*, abs/1710.07654, 2017.

[28] Kanishka Rao, Fuchun Peng, Haşim Sak, and Françoise Beaufays. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *Proc. of ICASSP*, 2015.

[29] M. Schroeder and B. Atal. Code-excited linear prediction(celp): High-quality speech at very low bit rates. In *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 937–940, 1985.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[31] Jürgen Trouvain and Fabian Brackhane. Wolfgang von kempelen's 'speaking machine' as an instrument for demonstration and research. In *International Congress of Phonetic Sciences*, 2011.

[32] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.

[33] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *CoRR*, abs/1606.05328, 2016.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[35] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: A fully end-to-end text-to-speech synthesis model. *CoRR*, abs/1703.10135, 2017.

[36] M Warmus and L Bole. Software implementation for odra 1204 of automatic phonemic transctiption of polish texts (in polish: Program na maszyne odra 1204 dla automatycznej transkrypeji fonematycznej tekstów języka polskiego). In *Zastosowanie maszyn matematycznych do badaú nad językiem naturalnym.* Wydawnictwo Uniwersytetu Warszawskiego, 1973.

[37] Xuesong Yang, Yun-Nung Chen, Dilek Hakkani-Tur, Paul Crook, Xiujun Li, Jianfeng Gao, and Li Deng. End-to-end joint learning of natural language understanding and dialogue manager, 2017.

[38] Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. Grapheme-to-phoneme conversion with convolutional neural networks. *Applied Sciences*, 9:1143, 03 2019.

[39] Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. Transformer based grapheme-to-phoneme conversion. In *Proc. of Interspeech*, 2019.

[40] Jian Zhu, Cong Zhang, and David Jurgens. Byt5 model for massively multilingual grapheme-to-phoneme conversion. pages 446–450, 09 2022.

[41] Markéta Řezáčková, Jan Švec, and Daniel Tihelka. T5G2P: Using Text-to-Text Transfer Transformer for Grapheme-to-Phoneme Conversion. In *Proc. Interspeech*, 2021.

Mirco Ravanelli suggested using curriculum learning to overcome the limitation of the initial model in processing long sequences, starting with the lexicon and then continuing on full sentences. Artem Ploujnikov expanded on the idea by adding a fine-tuning steps on the homograph dataset.

Mirco Ravanelli suggested using the CTC loss and tokenization to improve the model's performance. Artem Ploujnikov implemented and evaluated it.

First Article.

SoundChoice: Grapheme-to-Phoneme Processing

by

Artem Ploujnikov[1], and Mirco Ravanelli[2]

([1]) 3150 Rue Jean-Brillant, Montréal, QC H3T 1N8
Université de Montréal

([2]) 3150 Rue Jean-Brillant, Montréal, QC H3T 1N8
Université de Montréal

This article was submitted in INTERSPEECH 2022 (Accepted).

The main contributions of Artem Ploujnikov for this articles are presented.

- Model design (all additions to the basic encoder-decoder RNN baseline model originally included in SpeechBrain);
- Alternative Conformer-based model
- LibriG2P dataset preparation;
- Homograph loss;

- Mixed representations;
- Hyperparameter search using Orion
- Experiments;

Résumé. Les modèles de synthèse vocale de bout à bout visent à convertir une séquence de caractères directement dans une représentation de la voix, par exemple, un spectrogramme. Malgré leurs performances impressionnantes, de tels modèles ont de la difficulté à distinguer les prononciations des mots épelés identiquement, i.e. des homographes. Pour adresser ce problème, un modèle séparé graphème-phonème peut être utilisé pour convertir les caractères en phonèmes avant de synthétiser le son.

Cet article propose SoundChoice, une nouvelle architecture de modèle graphème-phonème qui peut traiter des phrases en entier au lieu de les traiter un mot à la fois. Cette méthode profite d'une fonction de perte d'homographes pondérée pour en faciliter la distinction, se sert de l'un programme d'apprentissage pour changer graduellement de la conversion d'un mot à la fois à la conversion d'une phrase entière, et elle se sert aussi des intégrations de mots BERT pour une amélioration de performance additionnelle. En plus, ce modèle inclut des meilleures pratiques adoptées de la reconnaissance de la parole, y inclus la classification temporelle connexionniste (CTC) et la recherche de faisceau avec un modèle de langage intégré. En conséquence, elle atteint un taux d'erreur de phonème de 2.65% en transcription des phrases entières avec des données de LibriSpeech et de Wikipédia.

Mots clés : graphème-phonème, synthèse vocale, phonétique, prononciation, disambiguation, homonymie

Abstract. End-to-end speech synthesis models directly convert the input characters into an audio representation (e.g., spectrograms). Despite their impressive performance, such models have difficulty disambiguating the pronunciations of identically spelled words. To mitigate this issue, a separate Grapheme-to-Phoneme (G2P) model can be employed to convert the characters into phonemes before synthesizing the audio.

This paper proposes *SoundChoice*, a novel G2P architecture that processes entire sentences rather than operating at the word level. The proposed architecture takes advantage of a weighted homograph loss (that improves disambiguation), exploits curriculum learning (that gradually switches from word-level to sentence-level G2P), and integrates word embeddings from BERT (for further performance improvement). Moreover, the model inherits the best practices in speech recognition, including multi-task learning with Connectionist Temporal Classification (CTC) and beam search with an embedded language model. As a result, SoundChoice achieves a Phoneme Error Rate (PER) of 2.65% on whole-sentence transcription using data from LibriSpeech and Wikipedia.

Keywords: grapheme-to-phoneme, speech synthesis, text-to-speech, phonetics, pronunciation, disambiguation

1. Introduction

Speech synthesis systems convert written text into a sequence of speech sounds. The irregularities commonly encountered in natural language orthography pose significant challenges to this process. For instance, a given sequence of characters (grapheme) can yield different pronunciations depending on the context (homographs). The sentence "English is t**ough** [ʌf]" can be understood thr**ough** [[uː] thor**ough** [ə] th**ough**t [ɑ] th**ough** [oʊ]. In some cases, the disambiguation depends on parts of speech (live - [laɪv] vs [lɪv]) or semantics (bass - [beɪs] vs [bæs]). Popular end-to-end speech synthesis models often fail to perform disambiguation of the homographs. Tacotron [**19**], for instance, is successful at only the most basic disambiguation (e.g. "read" - past vs present), while DeepVoice3 [**12**] produces intermediate phonemes in homographs.

Grapheme-to-Phoneme (G2P) models can improve the system's performance in these cases. Several approaches have been proposed in the literature: early attempts were mainly based on classical methods (e.g., Hidden Markov Models [**16**]), while more modern approaches rely on sequence-to-sequence deep learning. LSTM-based models [**13**] have been largely adopted for this task, and, more recently, transformer-based models [**22**] and convolutional models [**21**] have been proposed as well. These models are typically trained and evaluated on word-level lexicons (e.g., CMUDict [**40**]), making it impossible to resolve homograph disambiguation. The task of homograph disambiguation, on the other hand, has been explored in the literature as an independent research direction. Indeed, it was mainly framed as a classification task rather than an actual Grapheme-to-Phoneme conversion. Early work includes a classical hybrid method combining a rule-based algorithm and multinomial classifiers, such as the method proposed by Gornman et al., [**3**]. A BERT-derived classifier model based on contextual word embeddings [**10**] has been proposed as well. A recent example of a model exploiting sentence context is T5G2P [**30**]. DomainNet [**7**], instead, handles the task from a purely semantic view, while Alqahtani et al. [**10**] propose a self-supervised method for languages with diacritics that are frequently omitted.

This paper introduces *SoundChoice*, a novel G2P model that builds on insights from earlier contributions and addresses some of their prominent limitations. Different from previous methods, SoundChoice operates at the sentence level. This feature enables the model to exploit the context and better resolve homograph disambiguation. To further improve disambiguation, we propose a homograph loss that penalizes errors made on homograph words. The homograph disambiguation is not framed as a separate classification problem but is embedded into the G2P model itself through our homograph loss. In summary, the proposed SoundChoice introduces the following new features:

- It works at a sentence level, and it is trained with a weighted homograph loss.
- It gradually switches from word- to sentence-level G2P using a curriculum learning strategy.
- It models the sentence context by taking advantage of a mixed representation composed of characters and BERT word embeddings.
- It introduces Connectionist Temporal Classification (CTC) loss on top of the encoder and combines it with the standard sequence-to-sequence loss computed after the decoder (as commonly done in speech recognition).

Our best model achieves competitive Phonene-Error-Rate (PER%) on LibriSpeech sentence data (best test PER = 2.65%) with a homograph accuracy of 94%. The code[1] and the

[1] `https://github.com/speechbrain/speechbrain`

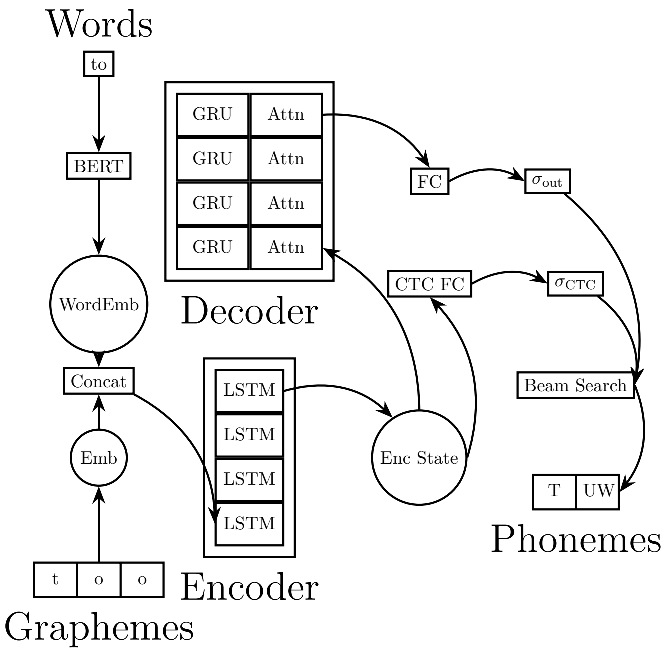


Fig. 1. Encoder-Decoder Architecture of SoundChoice.

pretrained model [2] are available on SpeechBrain [**15**]. We also release the new *LibriG2P* dataset that combines data from LibriSpeech Alignments [**8**] and the Wikipedia Homograph [**10**] on HuggingFace [3].

2. Model Architecture

The basic architecture of SoundChoice is depicted in Fig. 1. The input graphemes (discrete) are first encoded into continuous vectors using a simple lookup table that stores embeddings of a fixed dictionary and size. At this stage, we also combine word-level embeddings from a pretrained BERT model [**2**]. This addition inflates higher-level semantic information into the system that improves homograph disambiguation.

An LSTM-based encoder then scans the input characters and derives latent representations that embed short and long-term contextual information. On top of the encoder, we use a CTC loss (after applying a softmax classifier). The encoded states feed a GRU decoder coupled with a content-based attention mechanism. Special tokens called $\langle bos \rangle$ and $\langle eos \rangle$ are used to mark the beginning and end of a sentence, respectively. On top of the decoder, we combine the standard Negative Log-Likelihood (NLL) loss with our homograph loss. Finally, a hybrid beamsearch mechanism that exploits both the CTC and final predictions is employed. The partial hypotheses are rescored with an RNN language model that operates at the phoneme level.

[2] `https://huggingface.co/speechbrain`

[3] `https://huggingface.co/datasets/flexthink/librig2p-nostress-space`

We will provide more details on the proposed architecture in the following sub-sections.

2.1. Word Embeddings

To improve homograph disambiguation, we need our model to learn latent representations that correlate with grammar and semantics knowledge. We thus hypothesize that features from a large language model trained on a large corpus can improve performance. Although many of the recently-proposed language models could fit our purpose, we here used word embeddings derived from the popular BERT model [**2**]. The BERT embeddings pass through a simple encoder consisting of a normalization layer, a single downsampling linear layer, and tanh activation. These features are then concatenated with the character-level embeddings to form a single embedding vector.

2.2. Tokenization

We use the SpeechBrain [**15**] implementation of the SentencePiece [**18**] language-independent tokenizer with a unigram model. The goal is to shorten the grapheme and phoneme sequences, making them easier for the neural network to model. The tokenizer achieves this by learning a transformation to a newly constructed vocabulary comprised of the original tokens and common combinations of tokens encountered in the corpus. As we will discuss in Sec.5, we find that tokenization in the character and phoneme spaces is not always helpful and does not play an important role as expected.

2.3. Encoder/Decoder Architecture

The encoder and decoder use recurrent neural networks. The encoder is based on an LSTM, while the decoder uses a GRU model coupled with content-based attention[4]. The hyperparameters of the model are shown in Table 1. We derived them by performing a hyperparameter search with Oríon [**1**], where we search for the embedding dimensions, depths, and the number of neurons that maximize the PER on the validation set.

We attempted a variation of this model using residual convolutional layers [**21**]. In particular, we replaced the Bi-LSTM model with a series of residual convolutional layers. We achieve similar performance to the baseline one on lexicon data; however, it does not appear to benefit from pretraining, performing poorly on sentence data. We also conduct experiments to compare the RNN-based architecture with a Transformer-based one [**41**]. In this case, we use a conformer as an encoder and a standard transformer for decoding. The

[4] The choice of RNN model types is based on existing G2P and ASR models in SpeechBrain - using other combinations, such as both the encoder and decoder being an LSTM or a GRU is also possible; however, the comparison of different RNN architectures was not the primary focus of this work

Table 1. RNN Model Hyperparameters.

Component	Layer	Details
Embedding		Dim = 512
Encoder	LSTM	4 layers, 512 neurons, dropout = 0.5
Decoder	GRU	4 layers, 512 neurons, dropout = 0.5
FC	Linear	43 neurons
CTC FC	Linear	43 neurons

Table 2. Transformer (Conformer) Model Hyperparameters.

Component	Layer	Details
Embedding		Dim = 256
Encoder	Convolutional	2 layers, kernel size=15
Decoder	Transformer	2 layers, dim = 4096
FC	Linear	43 neurons
CTC FC	Linear	43 neurons

best hyperparameters are shown in Table 2. As we will see in Sec. 5, this model performs well but is slightly worse[5] than the aforementioned RNN-based architecture.

2.4. Beam Search and Language Model

We employ a hybrid beamsearcher similar to those used in modern speech recognizers [**20**]. It combines the log probabilities derived from the CTC encoder with those estimated by the decoder. The beamsearcher rescores the partial hypothesis with a phoneme language model. We hypothesize that a language model trained on sequences of phonemes can help minimize uncertainty by choosing the most likely phoneme sequences where an accurate prediction is difficult to make. We use an RNN-based language model with an embedding dimension of 256 and 2 hidden layers of 512 neurons each, regularized via dropout at a rate of 0.15.

3. Training

In this section, we provide more information on the adopted training strategy.

3.1. CTC Loss

The CTC[6] loss is computed on top of the encoder. CTC is suitable for grapheme-to-phoneme because the length of the phoneme sequence does not normally exceed the length of the input characters. This condition holds for many languages. For instance, in most

[5] See the Results section for likely reasons

[6] See the Introduction section for more details about CTC

European languages, a single grapheme can produce one phoneme by itself, be silent or be part of an n-graph. Languages producing more than one phoneme for single grapheme are rare. One exception is Ukrainian, where the letters "ϵ" and "ї" yield two-phoneme combinations [jɛ] and [jiː], respectively. In such cases, the limitation can be addressed via sequence padding or by introducing quasi-categories where a single position stands for two phonemes.

The CTC loss is combined with a standard NLL loss used on top of the decoder. This multi-task learning approach improves performance and helps the model convergence significantly.

3.2. Homograph Loss

In a typical ambiguous sentence from Wikipedia Homograph Data [**4**], the homograph only represents an insignificant portion of the whole sentence. An error in the homograph can involve only one or two phonemes out of 30-250 phonemes in the sentence. This incidence is comparable to random variations in labeling or infrequent, ambiguous, or challenging sequences, such as proper names or acronyms. A model can thus achieve a low PER without successfully disambiguating the homographs.

We mitigate this issue by adding a special loss that amplifies the contribution of the homographs relative to other words in the sentence. This is realized by computing the NLL loss on the subsequence corresponding to the homograph only.

The total loss used to train the G2P is thus a combination of three objectives:

$$\mathcal{L} = \mathcal{L}_{NLL} + \lambda_h \mathrm{L}_h + \lambda_c \mathrm{L}_{CTC} \tag{3.1}$$

where $\mathcal{L}_{NLL}$, $\mathcal{L}_h$, and L_{CTC} are the sequence, homograph, and CTC losses, respectively. The factors λ_h and λ_c are used to weight the homograph and CTC losses.

3.3. Curriculum Learning

We employ a curriculum learning strategy based on different stages of increasing complexity. First, we learn how to convert words into phonemes using the lexicon information. This step is relatively easy as it involves short sequences without the need to disambiguate homographs. In our case, we trained the model with this modality for 50 epochs. Then, we move training on by considering whole sentences from LibriSpeech-Alignments [**8**]. This step is more challenging, but the model pretrained on single words is already well-initialized for addressing this task. We train the model on sentences for 35 epochs. Finally, we perform a fine-tuning step using the homograph dataset for up to 50 epochs.

The adoption of this curriculum learning strategy turned out to play a crucial role in our G2P system. Without using it, the system provides a worse performance and struggles to converge. Further details on training can be found in the released code[7].

[7] `https://github.com/speechbrain/speechbrain/tree/develop/recipes/LibriSpeech/G2P`

Type	**Train**	**Validation**	**Test**	**Total**
Lexicon	202377	2065	2066	206508
Sentence	103967	2702	2702	109371
Homograph	9231	516	512	10259

Table 3. LibriG2P splits.

4. Experimental Setup

4.1. Datasets

We train the Grapheme-to-Phoneme Model using LibriSpeech-Alignments [**8**], Google Wikipedia Homograph Data [**3**] [**4**] and CMUDICT [**40**].

The set of outputs consists of 41 phonemes (ARPABET without stress markers) plus a word-separator token. The original phoneme annotations in LibriSpeech-Alignments [**8**] lack a word separator; its position is inferred from the word-level annotation. Google Wikipedia Homograph Data [**3**] [**4**], instead, lacks the phoneme annotations completely. However, each sample is tagged for the type of homograph it includes. Phoneme annotations are constructed by searching the tagged homograph in the provided glossary (with phonemes mapped from IPA to ARPABET) and looking up the remaining words in CMUDICT [**40**]. Uppercase words appearing in the original text that do not exist in CMUDICT [**40**] are interpreted as acronyms. We drop samples where the aforementioned methods fail.

We construct a new combined dataset named LibriG2P specialized for G2P with 3 slices: a *lexicon* consisting of each unique word encountered in LibriSpeech [**25**] as a separate sample, a *sentence* slice consisting of entire LibriSpeech dataset annotated for phonemes derived from Librispeech-Alignments [**8**] and a *homograph* slice consisting of a subset of the Wikipedia Homograph [**3**] [**4**] dataset. The non-space-enabled version lacks the homograph slice because the underlying implementation relies on word boundaries to locate the homograph. The train-validation-test split follows Table 3. The Google Wikipedia Homograph [**3**] [**4**] dataset is highly unbalanced with regards to the frequencies of pronunciation variations for any given homograph. We conduct experiments both with random sampling ("unbalanced") and with weighted sampling attempting to equalize the probability of each variation being selected ("balanced").

Given the inconsistency between LibriSpeech-Alignments [**25**] annotations obtained from audio and annotations computed using CMUDict [**40**], primarily in unstressed syllables or short connecting words - conjunctions and prepositions (e.g. "and": [ənd] vs [ænd] or into - [intuː] vs [intə], we produce a variation of the dataset with non-homograph words in the *sentence* slice replaced with CMUDict [**40**] pronunciations where possible.

To foster replicability and follow-up studies, we release the processed datasets to the community.

4.2. Metrics

We use the Phoneme Error Rate (PER%) to evaluate all models. To evaluate the performance of homograph disambiguation, we compute an additional metric, the *homograph classification accuracy*, defined as the percentage of samples in which the pronunciation of the homograph is predicted with no errors.

5. Results

5.1. RNN Model

Table 4 reports the performance achieved with the RNN model under different settings. It clearly emerges that sentence-based systems significantly outperforms word-based systems (see row 1 vs row 2 of Table 4). This change leads to a relative improvement of 47% in the PER, confirming the key importance of contexts in grapheme-to-phoneme conversion. Table 4 also highlights the importance of adding a special token (space) in the phoneme space. This token is needed to signal word boundaries and inject prior information about words into the system. This simple trick leads to a further 18% relative improvement of the PER. The tokenizer applied to phonemes leads to a minor performance improvement, while the BERT word embeddings do not improve the PER. BERT embeddings, however, will play a crucial role in homograph disambiguation. The phoneme language model turned out to not play a significant role as well. The best system achieves a PER of 2.65% on the LibriSpeech dataset.

5.2. Transformer Model

Table 5 reports the results achieved with the Conformer/Transformer model. The important benefits observed using a sentence-based system and adding the space token are confirmed. The minor role played by BERT embeddings and language models is observed for this model as well. In terms of performance, the best RNN model outperforms the transformer one (PER=2.65% vs PER=2.83%). The performance drop is not huge and might be because transformers notoriously require large datasets to be trained properly.

5.3. Homograph Disambiguation

Table 6 reports the results achieved after fine-tuning the model with the homograph dataset. The homograph disambiguation is evaluated on variations of the best-performing RNN model, with and without word embeddings. The use of the proposed homograph loss improves the homograph accuracy. With a weight factor λ_h of 2.0, the accuracy improves from 82% (no homograph loss) to 87%, thus corroborating our conjecture that the signal from the NLL loss on the entire sentence alone is not strong enough for disambiguation.

#	Sentence	Space	TP	WE	LM	Val PER	Test PER
1						6.82	6.46
2	✓					3.23	3.38
3	✓	✓				2.63	2.76
4	✓	✓	✓			2.56	2.69
5	✓	✓	✓	✓		**2.42**	2.71
6	✓	✓			✓	2.51	**2.65**

Table 4. G2P Model Results - RNN
Sentence is flagged when training/evaluating on full sentences, **Space** refers to the space token preserved. **TP** is marked when applying the tokenization to phonemes, **WE** refers to BERT embeddings, while **LM** is flagged when the phoneme language model is used.

#	Sentence	Space	TP	WE	Val PER	Test PER
1					9.11	9.23
2	✓				5.30	5.46
3	✓	✓			3.59	3.70
4	✓	✓	✓		**2.74**	**2.83**
5	✓	✓	✓	✓	2.79	2.97

Table 5. G2P Model Results - Conformer
Sentence is flagged when training/evaluating on full sentences, **Space** refers to the space token preserved. **TP** is marked when applying the tokenization to phonemes, **WE** refers to BERT embeddings, while **LM** is flagged when the phoneme language model is used.

BERT [**2**] embeddings significantly improve homograph detection as well. Thanks to this addition, the best system reaches an accuracy of 94% in homograph disambiguation. While the disambiguation accuracy cited in [**3**] is higher, this method achieves competitive results within the sequence model itself without additional classifiers.

It is worth mentioning that reevaluation of LibriSpeech [**25**] data after homograph fine-tuning showed a deterioration in nominal PER. This is due to inconsistencies in labeling, given that LibriSpeech Alignments was annotated using an automated aligner, capturing minor subtleties in pronunciation, whereas the homograph step relied on a dictionary [**40**] and allowed for only one pronunciation per word except for the homograph. When reevaluating on LibriSpeech Alignments after homograph fine-tuning, the test PER increases from 2.65% to 4.20%. Qualitative analysis reveals that most of the new errors originate from allowable variations in the labeling of non-homograph words, especially in prepositions/conjunctions and unstressed vowels. Retraining the model on the version of the dataset where the original labels are harmonized with CMUDict [**40**] leads to an overall PER decrease to 1.54%. This suggests that the apparent error increase in fine-tuning is due to a distribution shift rather than catastrophic forgetting.

	Word Emb	HG Weight	Bal	Accuracy
1		0.0		82 %
2		2.0		87 %
3		2.0	✓	85 %
4		5.0	✓	82 %
5	✓	2.0	✓	**94** %

Table 6. Homograph Disambiguation Results
Word Emb refers to Word embeddings, **HG Weight** is the weight of the homograph loss (λ_h), while **Bal** refers to balanced sampling.

6. Conclusions

This work proposed SoundChoice, a novel method for converting grapheme-to-phonemes that is robust against homograph disambiguation. The model is trained with a curriculum learning strategy that learns a word-based system first and finally learns a sentence-based model with a special homograph disambiguation loss. The best solution relies on an RNN system with hybrid/CTC attention and beam search. It takes advantage of word embeddings from a pre-trained BERT model as well. We achieved a PER of 2.65% on whole-sentence transcription using data from LibriSpeech and 94% accuracy in homograph detection using the Google Wikipedia Homograph [**3**] [**4**] corpus.

SoundChoice can be used in different ways in speech processing pipelines. For instance, it allows the training of TTS systems with phoneme tokens (or with mixed representation [**5**]). It can be used for speech recognition as well, as phonemes are known to be excellent targets, especially in channeling scenarios where speech is corrupted by noise and reverberation [**9**,**14**]. In future work, we would like to extend this approach to address multiple languages.

References

[1] Xavier Bouthillier, Christos Tsirigotis, François Corneau-Tremblay, Thomas Schweizer, Lin Dong, Pierre Delaunay, Mirko Bronzi, Dendi Suhubdy, Reyhane Askari, Michael Noukhovitch, Chao Xua, Satya Ortiz-Gagné, Olivier Breuleux, Arnaud Bergeron, Olexa Bilaniuk, Steven Bocco, Hadrien Bertrand, Guillaume Alain, Dmitriy Serdyuk, Peter Henderson, Pascal Lamblin, and Christopher Beckham. Epistimio/orion: Asynchronous Distributed Hyperparameter Optimization, May 2021.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, 2019.

[3] Kyle Gorman, Gleb Mazovetskiy, and Vitaly Nikolaev. Improving homograph disambiguation with supervised machine learning. In *Proc. of LREC*, 2018.

[4] Kyle Gorman, Gleb Mazovetskiy, and Vitaly Nikolaev. Homograph disambiguation data, July 2021.

[5] Kyle Kastner, João Felipe Santos, Yoshua Bengio, and Aaron Courville. Representation mixing for tts synthesis. In *Proc. of ICASSP*, 2019.

[6] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of EMNLP*, 2018.

[7] Aristotelis Leventidis, Laura Di Rocco, Wolfgang Gatterbauer, Renée J. Miller, and Mirek Riedewald. DomainNet: Homograph Detection for Data Lake Disambiguation. *EDBT 2021*.

[8] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. Speech Model Pre-Training for End-to-End Spoken Language Understanding. In *Proc. Interspeech*, 2019.

[9] Marco Matassoni, Ramón Fernandez Astudillo, Athanasios Katsamanis, and Mirco Ravanelli. The DIRHA-GRID corpus: baseline and tools for multi-room distant speech recognition using distributed microphones. In *Proc. of Interspeech*, 2014.

[10] Marco Nicolis and Viacheslav Klimkov. Homograph disambiguation with contextual word embeddings for tts systems. In *Proc. of Interspeech 2021 Workshop on Speech Synthesis (SSW11)*, 2021.

[11] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *Proc. of ICASSP*, 2015.

[12] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan Ömer Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep Voice 3: 2000-Speaker Neural Text-to-Speech. In *Proc. of ICLR*, 2018.

[13] Kanishka Rao, Fuchun Peng, Haşim Sak, and Françoise Beaufays. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *Proc. of ICASSP*, 2015.

[14] Mirco Ravanelli and Maurizio Omologo. On the selection of the impulse responses for distant-speech recognition based on contaminated speech training. In *Proc. of Interspeech*, 2014.

[15] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. SpeechBrain: A general-purpose speech toolkit, 2021. arXiv:2106.04624.

[16] Paul Taylor. Hidden markov models for grapheme to phoneme conversion. In *Proc. of Interspeech*, 2005.

[17] Carnegie Mellon University. CMU Pronouncing Dictionary, July 2021.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[19] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: A fully end-to-end text-to-speech synthesis model. *CoRR*, abs/1703.10135, 2017.

[20] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey, and Tomoki Hayashi. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253, 2017.

[21] Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. Grapheme-to-phoneme conversion with convolutional neural networks. *Applied Sciences*, 9:1143, 03 2019.

[22] Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. Transformer based grapheme-to-phoneme conversion. In *Proc. of Interspeech*, 2019.

[23] Markéta Řezáčková, Jan Švec, and Daniel Tihelka. T5G2P: Using Text-to-Text Transfer Transformer for Grapheme-to-Phoneme Conversion. In *Proc. Interspeech*, 2021.

Second Article.

MAdmixture: Multimodal Representations with a Common Embedding Space

by

Artem Ploujnikov[1], and Mirco Ravanelli[2]

([1]) 3150 Rue Jean-Brillant, Montréal, QC H3T 1N8
Université de Montréal

([2]) 3150 Rue Jean-Brillant, Montréal, QC H3T 1N8
Université de Montréal

This article was submitted in ICLR 2024 (in preparation).

The main contributions of Artem Ploujnikov for this articles are presented.

- The initial idea of applying the common latent space as seen in Tie Your Embeddings Down [**1**] to multimodal speech representation learning with a time dimension
- The Gaussian Mixture Time-Warping Aligner
- Training and evaluation of models
- Alignment ablation studies

Résumé. Ce travail vise à construire un modèle modulaire et unifié pour pouvoir exécuter des tâches diverses de traitement de la parole telles que la reconnaissance vocale, la synthèse texte-parole, la conversion graphème-phonème et d'autres tâches en se servant d'un réseau neural unifié modulaire au lieu d'entraîner un modèle séparé pour chacune des tâches. L'idée principale est d'encourager des modules du réseau à apprendre graduellement à partager un langage commun en se servant de représentations latentes similaires et alignées. En plus, le modèle peut servir comme un autoencoder pour chaque modalité. L'approche permet au modèle d'atteindre un niveau de performance comparable à celui des modèles spécifiques pour la conversion graphème-phonème, et il peut aussi apprendre la reconnaissance et la synthèse vocale comme preuve de concept. La méthode n'est pas limitée au traitement de la parole et elle peut être appliquée à des tâches diverses où des représentations d'une seule donnée dans plusieurs modalités peuvent être représentées de façon efficace dans un espace latent à l'aide d'un autoencoder et alignées l'une à l'autre dans la dimension temporelle. Elle se subit aussi à l'apprentissage par transfert où les encodeurs et les décodeurs pré-entraînés peuvent être intégrés dans le contexte multimodal avec d'autres tâches à apprendre. Cette approche introduit aussi un aligneur de déformation temporelle à mélange gaussien, une alternative au mécanisme traditionnel d'attention plus contraint, favorisant la stabilité d'entraînement.

Mots clés : multimodal, audio, speech, speech recognition, ASR, TTS, speech synthesis, grapheme-to-phoneme, G2P, latent, generative, diffusion

Abstract. This work aims to build a modular, unified approach to handling a range of speech processing tasks (speech recognition, text-to-speech, grapheme-to-phoneme and others) in a single modular network, rather than training a separate model for each. The core idea is to enable modality-specific sub-networks to progressively learn to speak a common language via shared, aligned latent representations. Additionally, the model learns an auto-encoder for each modality. Thus, a variety of generative approaches, such as latent diffusion, can be applied to generate samples in multiple modalities. The approach achieves a level of performance comparable to commonly used systems on grapheme-to-phoneme tasks, and is also able to learn ASR and TTS as a proof of concept. The method is not limited to speech and can be applied to a variety of multitask learning problems in which representations of a single data element in multiple modalities are temporally aligned and can be effectively represented in a latent space with an autoencoder. It is also amenable to transfer learning where encoders and decoders from existing pre-trained single-task models are integrated into the multimodal context with additional tasks to be learned. It introduces a novel method of aligning representations in the temporal dimension, the Gaussian Mixture Time-Warping Aligner, a highly constrained alternative to self-attention favouring training stability.

Keywords: multimodal, audio, parole, reconaissance vocale, synthèse vocale, graphème-phoneme, G2P, latent, génératif, diffusion

1. Introduction

The essence of a wide variety of artificial intelligence tasks can be thought of as transferring a signal or a piece of knowledge or information from one representation modality to another. Such tasks are particularly common in the area of speech and natural language processing where a given message can be written, spoken, handwritten or signed, as well as having the same content represented in different natural languages.

Speech processing is one of the oldest and best-studied applications of machine learning and artificial intelligence with a wide variety of practical uses in industry ranging from personal assistants to content search to automatic captioning and language translation. Typical tasks performed with speech processing include speech recognition [**12**, **20**](converting speech to written text), speech synthesis [**35**, **38**, **43**] (converting written text to speech), natural language understanding [**23**, **44**] (converting text and/or speech to actionable commands to be executed by an information system), speaker separation [**36**] (disentangling the overlapping speech of several speakers) and end-to-end translation [**16**]. Most such tasks involve studying and leveraging the interaction of representations of communications in human language across multiple modalities, including speech, written text, phonetic transcription, emotional labels or even raw video. For example, ASR is the transfer of a message from the audio modality to the text modality, whereas NLU is the transfer of a unit of meaning to a structured, machine-interpretable representation. For a given simples, modality representations differ in their richness and information content, and some express aspects of the message not found in others. Recorded speech represented as a waveform is the most high-dimensional and carries the semantic meaning of the message, as well as the speaker's voice timber, accent, intonation contours affected by the speaker's emotional state. A phonetic transcription of the speech carries only a subset of the information carried in raw speech, with phonemes representing only aspects of human speech used in distinguishing semantic units, discarding any variations in vocal characteristics resulting from speaker characteristics, intonation or emotion. Depending on the language used, written text may capture different subsets of this meaning. In languages with alphabetic writing systems, text corresponds closely to a phonetic transcription while carrying some information not encoded therein, such as punctuation, which may correlate with intonation, or homophones, representing different semantic units that require context and knowledge of the language to be disambiguated in speech or transcription. Languages with hieroglyphic writing systems may omit phonetic information entirely, as well as intonation, capturing only high-level semantics represented pictorially. Speech signal representation may also include secondary modalities, such as speaker identity/characteristics, emotions and intonation. These can, in theory, be derived from a waveform (as in speaker identification [**3**]) but cannot be used in isolation to reconstruct the signal in the primary modalities.

The present work studies the possibility of representing multiple modalities in speech in a single embedding space. Some prior work on common latent spaces has been explored in the literature. Examples include Amazon's Tie Your Embeddings Down [**1**] where a common latent space between audio and word embeddings was used for intent classification and SpeechT5 [**2**] where a common latent space was introduced with a specific model Transofrmer [**41**]-based architecture. We build on this prior work to devise a general approach. We also take advantage of the shared latent space to unconditionally generate samples in multiple modalities with a single generative sub-network. One example of a multimodal generative model using multiple autoencoders is multimodal latent diffusion [**32**]. Another recent work exploring the shared embedding idea is Meta AI's ImageBind [**10**]. However, the scope is different: it focuses mainly on single-vector embeddings of a single sample, lacking a time dimension, whereas MAdmixture learns sequential time-aligned representations. The concept of creating shared encoders for related tasks has also been explored in Unified Speech-Text Pre-training for

Speech Translation and Recognition [**39**]. However, like in [**2**], alignment is achieved using a combination of modality-specific and shared neural network modules. Existing methods are also available for time-series alignment; however, the have not been used extensively in the context of flexible multimodal architectures. One commonly used method is Dynamic Time Warping [**33**]; however, this original technique, as originally proposed, is not differentiable.

Compared to traditional single-task neural network TTS [**35**], ASR [**20**] and G2P [**30**] models, this approach makes it possible to reuse pre-trained subnetworks and the common latent space "language" they learn for adaptation to new modalities. Also, in order to facilitate content generation, a single generative model, such as latent diffusion [**5**], may be trained with the common latent representation as the generation target. Thus, a generated signal can then be decoded into multiple modalities. For models with 3 or more modalities, the approach is economical with respect to parameter count, requiring $2n$ networks to be trained (one encoder/aligner and one decoder per modality) plus n small aligner modules, compared to at least n^2 networks for the traditional non-multimodal approach.

2. MAdmixture Model

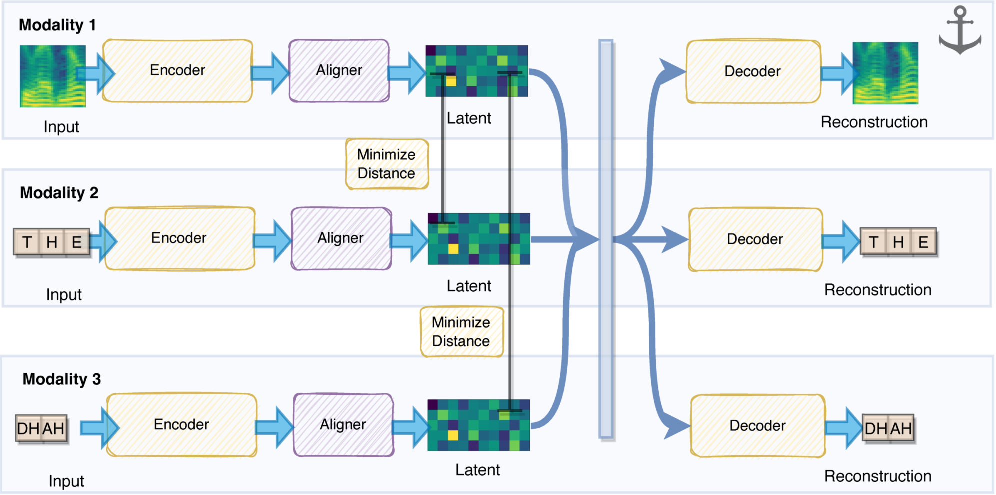


Fig. 2. MAdmixture Model Architecture

The present work introduces a novel audio model named MAdmixture (**M**odality **Admixture**) (shown in Figure 2), which aims to coalesce multiple modalities into a unified latent representation.

The proposed model introduces the following new features

- It introduces an approach where different modalities learn a common latent representation, facilitating the reuse of model components
- It introduces a new alignment technique, the Gaussian Mixture Time-Warping Aligner, which facilitates a learned dynamic scaling of latent representations

- It uses a combination of curriculum learning [**4**] and guided attention [**37**] to counter instabilities commonly encountered with multitask training, particularly those involving long sequences

The MAdmixture model consists of an arbitrary number of modalities representing the same signal. For each modality, the model contains:

- An **encoder** to convert the raw representation to a modality-specific latent representation
- An **aligner** to project the modality-specific representation to the common latent space approximately aligned in the time dimension
- A **decoder** to convert the common latent representation to a raw representation in a given modality.

The model architectures for the encoder and decoder are essentially arbitrary so long as they can learn good representations for the modality in question. Sub-networks from existing model architectures featuring an encoder and a decoder can be good candidates for integration into MAdmixture. For most speech modalities, RNN, Transformer [**41**] or convolutional encoders and decoders can be used. The model achieves all common tasks by extracting a latent representation from one modailty's encoder-aligner pair and then passing it to the decoder of the desired modality, which can be the same (autoencoding) or different (TTS, ASR, G2P, etc).

2.1. Aligner Architecture

The goal of the MAdmixture aligner is to learn a dynamic, variable time-warping function to scale the inputs in a given modality to roughly match the anchor modality in the time dimension. We hypothesize that while a self-attention mechanism is well-suited to computing an alignment between two sequences in the general case, for this specific problem, it is overly general, and a full attention mechanism is redundant given that most decoders that can be adapted from existing task-specific architectures already have an attention layer. Also, most current approaches to standard attention on variable-length sequences require target sequence lengths to be known in advance in order to avoid attending to zero padding. Early attempts at using based on traditional self-attention [**41**] suffered from too much instability for them to be practical. We thus re-frame alignment as a more constrained and explicit task of learning a time-warping function. This is achieved by predicting the width a given encoding vector occupies in the latent space irrespective of its absolute position. It is then reinterpreted as a distribution over positions in the latent space.

We propose a time-warping approach that is more constrained and stable than an attention mechanism with a strong prior of a near-monotonic, near-diagonal alignment where the weight of a given input in the latent space is defined by the Gaussian PDF. We use a simple linear layer to output the desired width of a given encoder output in the latent space.

The method transforms raw encoder outputs $\mathbf{X}^{(\mathrm{enc})}$ with a context-embedding transform $f^{(\mathrm{ctx})}$, such as a 1D convolution and then uses a width prediction network $f^{(w)}$ to predict the width and compute alignment matrix $\mathbf{A}$. An output projection $f^{(\mathrm{out})}$ and normalization $\mathrm{Norm}^{(\mathrm{out})}$ is then applied to obtain a final latent representation $\mathbf{X}^{(\mathrm{latent})}$ The aligner also outputs a length prediction $l^{(\mathrm{latent})}$, a simple sum of predictions for individual steps, to which a regression loss can then be applied during training. The method is shown in 1

Algorithm 1: Gaussian Mixture Time-Warping Aligner

```
$\mathbf{X}^{(\mathrm{ctx})} \leftarrow \mathrm{Norm}^{(\mathrm{out})}\left(f^{(\mathrm{ctx}}\left(\mathbf{X}^{(\mathrm{enc})}\right)\right)$ ;        /* Context embedding & input
 normalization */
$\hat{w}_t \leftarrow f^{(w)}(\mathbf{x}_t^{(ctx)})$ ;        /* Predict encoder time step latent width */
$\mu_t \leftarrow \sum_{t'=1}^{t-1} \hat{w}_{t'} + \frac{\hat{w}_t}{2}, \sigma_t \leftarrow \epsilon \frac{\hat{w}_t}{4}$ ;        /* Compute Gaussian prior mean/std */
$f^{(a)}(t', t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{t'-\mu_t}{\sigma_t}\right)^2\right)$ ;        /* Gaussian prior */
$\mathbf{A}_{t',t} \leftarrow \frac{f^{(a)}(t',t)}{\sum_{k=1}^{l} f^{(a)}(t',k)}$ ;        /* Normalized alignment matrix */
$\mathbf{X}^{(\mathrm{latent})} \leftarrow \mathrm{Norm}^{(out)}\left(f^{(\mathrm{out})}\left(\mathbf{X}^{(\mathrm{ctx})}\mathbf{A}^\top\right)\right)$ ;  /* Aligned & normalized outputs */
$l^{(\mathrm{latent})} \leftarrow \sum_{t=1}^{l} \hat{w}_t$ ;        /* Latent length prediction */
```

Unlike methods based on attention, such as the classic self-attention method used in Transformers [**41**], this alignment method is naturally diagonal and not prone to the common training instabilities encountered with classic attention (e.g. failure to align). To address possible numerical instabilities when length predictions are close to zero, we use a scale-and-shift technique described in detail in Appendix 8.1.

2.1.1. Model Architecture. The baseline model is based on the CRDNN (convolutional + recurrent + deep/fully connected neural network) ASR found in SpeechBrain [**29**] and SoundChoice G2P [**28**] with some hyperparameter adjustments made to accommodate a common latent space. For detailed hyperparameter settings used in the model, refer to Appendix 8.2.

We also attempt a preliminary proof-of-concept where the CRDNN encoders and decoders are replaced with their Conformer [**13**]/Transformer [**41**] counterparts (see appendix 8.7).

2.2. Sampling

Generative models based on latent diffusion [**5**] have been shown in literature to achieve state-of-the-art results on a variety of tasks, most notably image generation. We use a similar approach to generate audio and text samples simultaneously by training a residual UNet [**31**]. We use channel multipliers of 1 and 2 with 2 residual layers per multiplier. The UNet [**31**] implementation from OpenAI Guided Diffusion [**24**] was used as a base for the UNet [**31**], and Phil Wang's open-source *denoising-diffusion-pytorch* [**42**] was used as a guide to implementing diffusion within SpeechBrain.

On each training step, we randomly choose a latent representation from one of the available modalities and then add a certain amount of random noise sampled from a time step schedule and train the UNet to predict the noise as suggested in [**15**]. This process is then reversed during influence to produce a latent sample, which can then be fed to multiple decoders to obtain generated signals in multiple modalities. The sampler is used to obtain generated signal samples across multiple modalities (e.g. of speech, text, etc).

3. Theoretical Problem Formulation

3.1. The Traditional Approach

In the traditional approach where a single model is trained for a specific task, such as ASR, TTS or G2P, the model transfers from the source modality to the target modality:

$$\hat{\mathbf{x}}^{(\mathrm{tgt})} = f_{\mathrm{model}}(\mathbf{x}^{(\mathrm{src})})$$

$\hat{\mathbf{x}}^{(\mathrm{tgt})}$ is the predicted signal in the target modality, and $\mathbf{x}^{(\mathrm{src})}$ is the signal in the source modality. For example, in the case of ASR, $\hat{\mathbf{x}}^{(\mathrm{tgt})}$ is the predicted text sequence (e.g. a sequence of characters or tokens), whereas $\mathbf{x}^{(\mathrm{src})}$ is the source audio signal.

The problem can thus be framed as finding the message representation in the target modality with the highest probability given the message in the source modality.

$$\hat{\mathbf{x}}^{(\mathrm{tgt})} = \underset{\mathbf{x}^{(\mathrm{tgt})} \sim \mathcal{X}^{(\mathrm{tgt})}}{\arg\max}\ \mathbb{P}\left(\mathbf{x}^{(\mathrm{tgt})} | \mathbf{x}^{(\mathrm{src})}\right)$$

Consider the most common applications: ASR and text-to-speech. In the case of ASR, $\mathbf{x}^{(\mathrm{src})} = \mathbf{x}^{(\mathrm{audio})}, \mathbf{x}^{(\mathrm{tgt})} = \mathbf{x}^{(\mathrm{text})}$. In the case of TTS, $\mathbf{x}^{(\mathrm{src})} = \mathbf{x}^{(\mathrm{text})}, \mathbf{x}^{(\mathrm{tgt})} = \mathbf{x}^{(\mathrm{audio})}$. Other common applications can be formulated similarly. For instance, in emotion recognition, $\mathbf{x}^{(\mathrm{src})} = \mathbf{x}^{(\mathrm{audio})}, \mathbf{x}^{(\mathrm{tgt})} = \mathbf{x}^{(\mathrm{emotion})}$.

3.2. Extension to Multimodal Processing

In some cases, one may consider extending this to multiple modalities. For instance, one may want to synthesize speech expressed with a specific emotion in cases where the emotional content cannot be unambiguously inferred from the textual representation.

The formulation then becomes

$$\left(\hat{\mathbf{x}}_1^{(\text{tgt})}, \hat{\mathbf{x}}_2^{(\text{tgt})}, ..., \hat{\mathbf{x}}_k^{(\text{tgt})}\right) = \underset{\substack{\mathbf{x}_1^{(\text{tgt})} \sim \mathcal{X}_1^{(\text{tgt})} \\ \mathbf{x}_2^{(\text{tgt})} \sim \mathcal{X}_2^{(\text{tgt})} \\ ... \\ \mathbf{x}_k^{(\text{tgt})} \sim \mathcal{X}_k^{(\text{tgt})}}}{\arg\max} \mathbb{P}\left(\mathbf{x}_1^{(\text{tgt})}, \mathbf{x}_2^{(\text{tgt})}, ..., \mathbf{x}_k^{(\text{tgt})} | \mathbf{x}_1^{(\text{src})}, \mathbf{x}_2^{(\text{src})}, ..., \mathbf{x}_j^{(\text{src})}\right)$$

Thus, the model needs to find the most likely combination of samples from the chosen k target modalities given specific samples from j source modalities.

Commonly deployed speech models do not currently address this problem generically. However, specific model architectures performing this for select modalities are available. For instance, multispeaker TTS systems, such as DeepVoice3 [**27**], combine text representations with speaker embeddings to produce speech in the desired voice.

3.3. Common Latent Space Embeddings

The proposed approach involves a joint training approach, which would remove the explicit distinction between source and target modalities. Instead, let $\mathbf{x}^{(m)}$ be the representation of a sample in modality m. Additionally, instead of using a task-specific pre-processing function, each modality will have an encoder function that produces an embedding of the signal in that modality and a decoder function $\hat{f}^{-1}(\mathbf{z}^{(m)})$ that will attempt an approximate reconstruction from the common embedding space into that same modality, with error $\epsilon^{(m)}$. The approach is inspired by Tie Your Embeddings Down [**1**].

$$\mathbf{z}^{(m)} = f_{\text{enc}}^{(m)}(\mathbf{x}^{(m)}), \mathbf{x}^{(m)} = \hat{f}_{\text{enc}}^{-1(m)}(\mathbf{x}^{(m)}) + \epsilon^{(m)}$$

Intuitively, we want to learn a transformation of all available modalities into the shared embedding space, such that:

- The reconstruction error within a single modality is minimized (i.e. a faithful reconstruction of any modality can be obtained from the latent representation)
- For a given speech sample, the projections of all modalities into the latent space are closer to each other than they would be to the projection of any modality from an unrelated, differently-sounding speech sample
- The time dimension in the latent space is shared across modalities
- The time dimension of the latent space embedding is preserved, at least approximately. That is, given a latent representation $\mathbf{Z} \in \mathbb{R}^{t_z \times f_z}$ of raw input $\mathbf{X} \in \mathbb{R}^{t \times f}$, and for $k_z < t_z$, one were to construct a subset $\mathbf{Z}' = \mathbf{Z}_{0:k_z,f}$, one can approximately reconstruct the first $\frac{k_z t}{t_z}$ time steps on the input with no access to the rest of the latent space tensor. Here t_z refers to the number of time steps in the latent space f_z refers to the size of the latent feature dimension.

Examples of modality transfers include speech recognition (audio → text), phonetic transcription (audio → phonemes), speech synthesis (text → audio, phonemes → audio), G2P (text → phonemes).

More formally, the objectives can be formulated as follows:

$$\begin{aligned}
&\min \sum_{m \in M} \mathbb{E}\left[d(\left(\hat{\mathbf{x}}^{(m)}, \mathbf{x}^{(m)}\right))\right] = \min \sum_{m \in M} \mathbb{E}\left[d\left(\hat{f}_{\text{enc}}^{-1(m)}(\hat{f}_{\text{enc}}^{(m)}(\mathbf{x}^{(x)})), \mathbf{x}^{(m)}\right)\right] \\
&\min \sum_{m \in M} \mathbb{E}\left[d(\mathbf{z}^{(m)}, \bar{\mathbf{z}})\right]
\end{aligned}$$

where $d(\mathbf{x}_1, \mathbf{x}_2)$ is a distance metric between vectors or tensors (e.g. MSE, cosine distance)

3.4. Sequence Length Handling

The task of mapping multiple modalities to a common latent space involves finding a common representation of the time dimension. This task is not straightforward because the representation of a given segment in different modalities can vary significantly in length. Additionally, for some modalities, such as raw characters, the mapping is not entirely equivariant with many orthographies relying on polygraphs to represent a single phoneme, as well as featuring silent letters that have no representation in the phoneme or audio modalities at all.

For certain tasks in prior work, such as intent classification in Tie Your Embeddings Down [**1**], the time dimension can be discarded completely. However, in the task featured in the present work, such as TTS, ASR and G2P, we consider it useful to preserve the time dimension.

The *aligner*: $\mathbf{z}_t^{(m)} = A_t^{(m)}(\mathbf{x}_1^{(m)}, \mathbf{x}_2^{(m)}, ..., \mathbf{x}_n^{(m)})$ is used to map a raw encoded representation from a single modality to the time dimension. For practical reasons, we assume that a desirable latent representation is *approximately equivariant* in the time dimension. While it is not necessary to satisfy this assumption in order to achieve good performance on the tasks being considered, this property is useful for interpretability and modularity.

MAdmixture addresses the problem of temporal representation alignment by choosing an anchor modality to which all modalities aligned, i.e. $\forall m \in M, \mathbb{E}[l^{(m)}] = l^{(\text{anchor})}$. We handle this by learning a length prediction $f^{(l)}(\mathbf{X}^{(enc)})$ so as to minimize the difference between the predicted length and ground truth anchor lengths $\arg\min_{\mathbf{\Theta}} \mathbb{E}[f^{(l)}(\mathbf{X}^{(\text{enc})}) - l^{(\text{anchor})}]$

3.5. Sampling

With the common latent representation approach, generating samples across modalities can be reduced to sampling from a single latent distribution $\mathbf{z} \sim Z$, which can be parameterized using a single neural network.

3.6. Loss Function

The proposed optimization criterion is:

$$\begin{aligned}\mathcal{L} = \lambda_{\text{rec}}\,\mathcal{L}_{\text{rec}} + \lambda_{\text{transfer}}\,\mathcal{L}_{\text{transfer}} + \lambda_{\text{distance}}\,\mathcal{L}_{\text{distance}} + \lambda_{\text{align}}\,\mathcal{L}_{\text{align}} + \lambda_{\text{len}}\,\mathcal{L}_{\text{len}} \\ + \lambda_{\text{ctx}}\,\mathcal{L}_{\text{ctx}} + \lambda_{\text{sampler}}\,\mathcal{L}_{\text{sampler}}\end{aligned}$$

$\mathcal{L}$ is the total loss, $\mathcal{L}_{\text{align}}$ is the alignment loss $\mathcal{L}_{\text{rec}}, \lambda_{\text{rec}}$ are the reconstruction loss and its weight, $\mathcal{L}_{\text{transfer}}, \lambda_{\text{transfer}}$ are the transfer loss and its weight, $\mathcal{L}_{\text{distance}}, \lambda_{\text{distance}}$ are the latent space distance loss and its weight $\mathcal{L}_{\text{len}}, \lambda_{\text{len}}$ are the length loss and its weight, $\mathcal{L}_{\text{ctx}}, \lambda_{\text{ctx}}$ are the context loss and its weight and $\mathcal{L}_{\text{sampler}}, \lambda_{\text{sampler}}$ are the sampler loss and its weight.

The reconstruction loss $\mathcal{L}_{\text{rec}} = \sum_{m \in M} d^{(m)}(\bar{\mathbf{x}}^{(m)}, \mathbf{x}^{(m)})$ is expected to be modality-specific, appropriate for the model being used. For the baseline model, we use the MSE loss for spectrograms and the Negative Likelihood Loss for character modalities. The latent space distance loss $\mathcal{L}_{\text{dist}}$ can encourages the representations of different modalities from the same example to be closely aligned. The simplest approach involves computing the stepwise MSE loss between the latent representation in the anchor modality and other modalities. More complex contrastive losses can also be used.

The purpose of the alignment loss is to improve training stability in any attention-based decoders. The approach used in the present work involves applying a guided attention loss as proposed in Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention [**37**]

$$_{k_{\text{in}},k_{\text{out}}} = 1 - \exp\left(\frac{k_{\text{in}}}{L_{\text{in}}} - \frac{k_{\text{out}}}{L_{\text{out}}}\right) \mathcal{L}_{\text{align}} = \mathbb{E}\left[_{k_{\text{in}},k_{\text{out}}}\,_{k_{\text{in}},k_{\text{out}}}\right]$$

The transfer loss ensures that a decoder of a given modality can effectively reconstruct latent representations from the encoders of other modalities.

$$\mathcal{L}_{\text{transfer}} = \sum_{\text{src} \in M, \text{tgt} \in M, \text{src} \neq \text{tgt}} d^{(\text{tgt})}(\text{Dec}^{(\text{tgt})}\left(\text{Align}^{(\text{src})}\left(\text{Enc}^{(\text{src})}(\mathbf{x}^{(\text{src})})\right)\right), \mathbf{x}^{(\text{tgt})})$$

$d^{(tgt)}$ is the modality-specific distance function between target outputs and ground truths, such as MSE, negative log-likelihood, etc.

The context loss $\mathcal{L}_{\text{ctx}}$ refers to any modality-specific losses applied to decoders. In the baseline implementation, we use a CTC [**11**] loss to improve decoding for character and phoneme modalities and a Tacotron [**35**] gate loss to predict sequence lengths.

The sampler loss $\mathcal{L}_{\text{sampler}}$ refers to the loss function applied to the sampler network for the generative context. For a Denoising Diffusion [**15**] sampler, this is the MSE loss between the added noise and the predicted noise.

4. Related Work

The idea of creating a common embedding space for multiple languages or modalities is inspired by the Tie Your Embeddings Down [**1**] work from Amazon, which combines pretrained sentence embeddings and raw audio embeddings in a single embedding space aiming to improve the performance of Natural Language Understanding tasks. Outside of speech processing, the idea has been explored for word embeddings in Learning Multilingual Word Embeddings in Latent Metric Space: A Geometric Approach where a language-specific embedding is learned as a rotation of the original embedding. The idea has also been applied to language translation [**8**] and sentence embeddings [**9**].

Similar ideas have been explored in the speech context SpeechT5 [**2**] model. However, whereas SpeechT5 proposes a very concrete architecture for a joint TTS and ASR model, the present model proposes a meta-learning technique that can be used with a variety of sequence models, such as GRU [**6**] or other recurrent models, Transformer [**41**]-based. The main advantage of our proposed approach is that rather than proposing a specific architecture for a specific task, we propose a general technique in which a base model can be trained and then easily expanded further to other modalities.

While the latent space alignment approach proposed in this work has been designed specifically for this multimodal learning task, similar concepts of using Gaussian priors for dynamic time-warping have been proposed in Gaussian Process Latent Variable Alignment Learning [**17**]. However, it introduces a more complex statistical model for direct learning on sequences, whereas our proposed approach functions more as a highly constrained scaled self-attention mechanism.

In recent years, a lot of research has been focusing on both conditioned and unconditioned generation of content in different modalities with a major body of literature focusing on conditioned image generation. One popular method is denoising diffusion [**15**] where a model is trained to predict Gaussian noise given a noisy sample, and the process is reversed stepwise during inference, starting from pure noise and removing it one step at a time to generate a sample. Our method incorporates denoising diffusion to generate new samples in the available modalities.

5. Results

5.1. Training

Two data sets are being used for the common latent space model : LibriSpeech [**25**] is an multi-speaker audio dataset comprised of 1,000 hours of read English speech and LibriSpeech-Alignments [**22**] enhances the LibriSpeech [**25**] dataset with explicit phoneme alignment annotations. The model does not require alignments for training unless the underlying TTS does, we only use LibriSpeech-Alignments [**22**] for phoneme modality labels. To facilitate curriculum learning, we use a curriculum of gradually increasing sequence length described in Appendix 8.3. We also attempt to enhance ASR performance by applying SpecAugment [**26**], which aims to remove portions of the spectrogram during training.

5.2. Evaluation

Since the main practical purpose of this model is to transfer the representation of a signal from one modality to another, the most important evaluation metric is the accuracy of reconstruction, for each modality. For tasks involving sequential target modalities, including audio to character, audio to phoneme and character to phoneme using the Token Error Rate metric based on the Levenshtein edit distance [**19**]: $TER = \frac{n_{\text{ins}}+n_{\text{del}}+n_{\text{sub}}}{n}$, on previously unseen data.

We conduct a preliminary experiment where curriculum learning is disabled, but find initial convergence too slow and unstable for the approach to be practical.

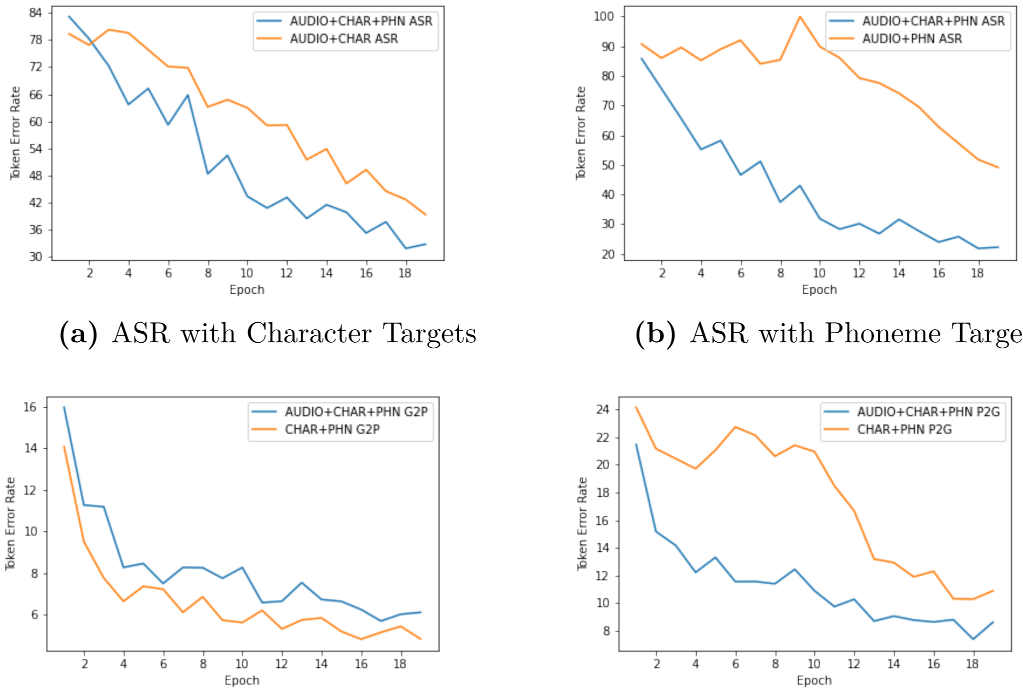
To evaluate the effectiveness of the Gaussian Mixture Aligner, we compare the training trajectory of a phoneme-grapheme MAdmixture that uses the aligner to that of a model that uses a simple projection instead. We choose a model with text modalities only because they can be evaluated using a simple objective metric (Token Error Rate). The evaluation details are described in appendix 8.1.

5.3. Results

Table 7 summarizes the results on modality transfer tasks that were evaluated. The Data column evaluates the type of data used for evaluation: *Segment* indicates that sampled segments from LibriSpeech (clean) of up to 20 words are used in curriculum learning, *Full* indicates complete examples from LibriSpeech (clean). *SA*indicates that SpecAugment [**26**] and non-clean data is used.

Model	Data	aud - phn	aud - char	char - phn	phn - char
MM: AUDIO + CHAR	Segment	—	3.8%	—	—
MM: AUDIO + PHN	Segment	—	4.3%	—	—
MM: AUDIO + CHAR + PHN	Segment	4.2%	4.0 %	3.2 %	2.2 %
MM: CHAR + PHN	Segment	—	—-	3.3%	2.2 %
MM: AUDIO + CHAR	Full	—	4.1%	—	—
MM: AUDIO + CHAR SA	Full	—	3.2%	—	—
ASR - SpeechBrain - CRDNN	Full	—	1.2 %	—	—
ASR - SpeechBrain - Transformer	Full	—	2.3 % (WER)	—	—
ASR - SpeechT5 - Transformer	Full	—	1.9 % (WER)	—	—
G2P - SoundChoice - RNN	Full	—	—	2.7 %	—
G2P - SoundChoice - Conformer	Full	—	—	2.8 %	—
G2P - Conv[8]	Word (CMU)	—	—	4.8 %	—

Table 7. MAdmixture Performance Comparison
All metrics are Token Error Rates (phoneme or character) - except where indicated using (WER) where only a Word Error Rate (WER) is available



(a) ASR with Character Targets

(b) ASR with Phoneme Targets

(c) Grapheme-to-Phoneme

(d) Phoneme-to-Grapheme

Fig. 3. Multimodal Learning Effect Comparison

We compare the training trajectory of a model with three modalities (audio, character and phonemes) with different combinations of two-modality models (character + phoneme, audio + phoneme, audio + character) in order to determine the marginal effect of adding a modality. We find empirically that at least during early training stages, training on more modalities leads to better performance of a given modality transfer, and the effect is observed across multiple modality combinations. We hypothesize that given the approach relies on shared representations, learning encoder representations and latent alignments for a given modality improves performance in other modalities. The effect is less noticeable during later stages of training or when using aggressive gradient clipping.

We find that G2P performance is similar to that of the single-task SoundChoice G2P [**28**] trained and evaluated on the same dataset - LibriSpeech [**25**] with LibriSpeech-Alignments [**22**].

The remaining gap between MAdmixture results and single-task models of comparable capacity for ASR could be due to the fact that the best results were achieved with SentencePiece [**18**] tokenization and a beam search trained with a large language model. With MAdmixture, preliminary experiments with tokenization produced subpar results compared

[8] Grapheme-to-Phoneme Conversion with Convolutional Neural Networks, evaluated on CMUDict [**40**]

to raw character embeddings, which is likely due to a greater difference in length between the representations that need to be aligned.

For speech synthesis, we observe that as expected, for a given training stage the quality of the audio autoencoding reconstruction is superior to reconstructions from token sequence modalities. We also note that given that we train a single-speaker TTS on multispeaker data, over time, reconstructions converge to an "average" voice, and this affects both autoencoding reconstructions and reconstructions from text.

We conclude that the proposed approach is capable of effectively supporting common speech processing tasks: ASR, TTS, G2P, and extending it to others can be further evaluated.

6. Limitations

The model created in this work are limited to a single language and a single dataset [**25**]. It is unclear how well it would transfer to other languages, particularly those with non-alphabetic writing systems. Despite being generic in theory, the evaluation is limited to only a single family of architectures (RNN/CRDNN) plus a preliminary Conformer [**13**] experiment and to only three modalities (audio, characters, phonemes). Further study is needed on the quality/transferability of representations learned and how they scale with data and/or compute compared to other approaches. It is also unclear whether the specific alignment methodology used in these experiments can be easily extended to modalities with a very high variance in time dimension alignment.

7. Future Work

Future research can build on the present work to achieve state-of-the-art performance in the modalities explored in the present work with various techniques used in state-of-the-art TTS/ASR models, including data augmentation and artificial noise.

One possible line of follow-up work is to expand the approach to other modalities representing human speech and language, such as lip reading, sign language or handwriting. The approach can also be used with other neural architectures.

Further empirical research is also needed on the scaling properties of such models with respect to compute and data and how they compare to traditional approaches, possibly deriving scaling laws [**14**].

8. Appendix

8.1. Alignment

8.1.1. Gaussian Mixture Time-Warping Aligner. We aim to design an aligner such that

- On average, the modality inputs are scaled by a constant *scale* (set by a hyperparameter) when projected to the latent space
- A given input feature is assigned a specific width in the latent space
- The widths can very continuously, and this can be accounted for in a representation of fixed resolution similar to how subpixel interpolation is used in image scaling

The built-in prior of the proposed aligner is that the distribution of time positions in the output (latent) space corresponding to a given encoded position corresponds to Gaussian distribution centered at the midpoint of the cumulative predicted width.

By default, the standard deviation is set so that the predicted width corresponds to 4 standard deviations, putting 95 percent of the total probability density within the segment of width $\hat{w}_t$. We introduce a *spread* hyperparameter ϵ, which makes it possible to adjust the distribution, increasing or degree to which a given input step blends into the position in the latent space occupied by previous and following steps.

$$\mu_t = \sum_{t'=1}^{t-1} w_{t'} + \frac{w_t}{2}$$

$$\sigma_t = \epsilon \frac{w_t^{(pred)}}{4}$$

We attempt to incorporate some immediate context information for the encoder representations to ensure the aligner can effectively estimate width in situations where the alignment is dependent on neighboring tokens or steps. For instance, in the case of graphemes, this is the case with n-graphs. To this end, we propose a context transformation, such as a series of 1-D convolutions. By default, we use a single convolutional layer to compute the context to which the width prediction function is then applied.

$$\mathbf{X}^{(\text{ctx})} = \text{Norm}^{(in)}(f^{(ctx)}(\mathbf{X}^{(enc)}))$$

$$\hat{w}_t = f^{(w)}\left(\mathbf{X}^{(\text{ctx})}\right)$$

By default, we use a single linear layer on the context outputs as a width predictor with sigmoid activation, for simplicity, outputting a value of up to $w_{\max} = 2s$. In preliminary experiments, we find that allowing the predictor to output very small lengths causes numerical instability, which we address by constraining the range of the activation output. An additional *sensitivity* hyperparameter is used to prevent small perturbations from saturating the sigmoid activation.

$$\begin{aligned}
\mathbb{E}[f^{(w)}(\mathbf{x}_t^{(ctx)})] &\approx s \\
\forall \mathbf{x}_t^{(ctx)} f^{(w)}(\mathbf{x}_t^{(ctx)}) &\in (w_{\min}, w_{\max}) \\
\mathbf{o}_t^{(w)} &= \mathbf{W}^{(w)}\mathbf{x}_t^{(ctx)} + \mathbf{b}^{(w)} \\
\hat{\mathbf{w}}_t = f^{(w)}(\mathbf{X}^{(ctx)}) &= w_{\min} + (w_{\max} - w_{\min})\sigma\left(\gamma + \beta \mathbf{o}_t\right)
\end{aligned}$$

We want to choose a value γ such that for typical initializations where the expected of the linear layer's output is zero, the expected output is still s despite the shift by $w_{\min}$. We thus choose the following value:

$$\gamma = \sigma^{-1}\left(\frac{s - w_{\min}}{w_{\max} - w_{\min}}\right)$$

This relies on an assumption that $\mathbb{E}[\sigma(x)] \approx \sigma(\mathbb{E}[x])$, which, while not generally true, holds in the portion of the sigmoid function that is approximately linear.

We subsequently set up the Gaussian PDF prior for the alignments as follows, where t is the input step and t' is the output step.

$$f^{(a)}(t', t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{t' - \mu_t}{\sigma_t}\right)^2\right)$$

We then collect the discrete approximations of the individual Gaussians into a matrix and normalize it over the inputs so that for any given output the contributions of inputs add up to one to obtain an alignment matrix $\mathbf{A}$ similar to one used in a standard attention mechanism. Similarly, aligned latent representations can then be obtained by multiplying context-embedded input by the alignments and then applying an output projection $f^{(out)}$ and an output normalization $\text{Norm}^{(\text{out})}$

$$\begin{aligned}
\mathbf{A}_{t',t} &= \frac{f^{(a)}(t', t)}{\sum_{k=1}^{l} f^{(a)}(t', k)} \\
\mathbf{X}^{(\text{latent})} \in \mathbb{R}^{d^{(\text{latent})} \times l'} &= \text{Norm}^{(out)}\left(f^{(\text{out})}\left(\mathbf{X}^{(\text{ctx})}\mathbf{A}^{\top}\right)\right)
\end{aligned}$$

8.1.2. Alignment Hyperparameters. Table 8 shows the hyperparameters used for the modalities studied. We observe that the distribution of length ratios is approximately normal with a "long tail" of outliers. We determine the appropriate scale value by first setting the scale of the anchor modality to 1.0 and then determining the average ratio of encoded representations in the train set, excluding any values above the 90% percentile. For most modalities, we observe that the system learns alignments as expected achieving adequate performance with the default spread value of 1.0 with the exceptions of pretrained experiments with tokenization[9] where a qualitative examination reveals gaps in the latent space. As a result, we increase the value of the spread hyperparameter slightly.

Modality	Scale	Spread
AUDIO	1.0	1.0
CHAR	1.7	1.0
PHN	2.5	1.0
CHAR - Tokenized	5.2	1.5

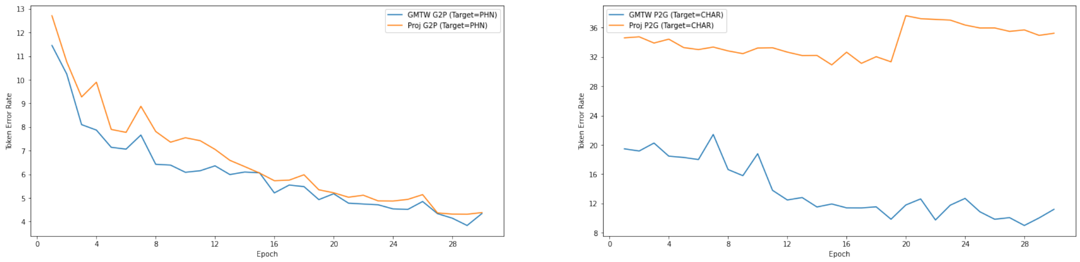
Table 8. MAdmixture Alignment Hyperparameters

8.1.3. Evaluation. The raw encoded output length distributions and the distribution of latent representation lengths predicted by the aligner is shown in Table 9 and Figure 5. At face value, the distribution confirms that the aligner set-up combined with the alignment loss encourages the samples to be scaled, on average, to match the ground truth, i.e. the length of the raw encoder outputs in the anchor modality. Figure 6 shows the distributions of alignment errors (i.e. absolute differences between the aligner-assigned total length of the sample and the ground truth) and compares it to the error obtained if one were to naively, deterministically scale the output by the value of the scale hyperparameter. We observe a reduction in median error from 11.71 to 9.69 for characters and from 12.07 to 8.80 for phonemes. One should note that the aligner loss is combined with the other losses in a way where reconstruction and transfer losses dominate over the alignment loss, which discourages the model from learning a better alignment if its impact on decoding is marginal.

One notable finding regarding the distributions of both lengths and errors is the large number of outliers. A qualitative error analysis revealed that particularly unusual length ratios are typically due to the presence of pauses in the recordings. A possible improvement to this aligner architecture that could be considered in future work is the ability to explicitly account for positions in the latent space for which the encoder in a given modality does not have a corresponding output position (i.e. the ability to "skip" positions). Another possible alternative is data pre-processing to remove long pauses.

The results of the ablation experiment for the Gaussian Mixture Time Warping Aligner are shown in Figure 4. We observe faster convergence for both modalities when the aligner is used - compared to the model that only using a simple projection, preserving the lengths from individual modality encoders. The observed effect is stronger for phoneme-to-character conversion than for character-to-phoneme conversion. We hypothesize that this effect is likely due to the aligner producing more similar representations across modalities.

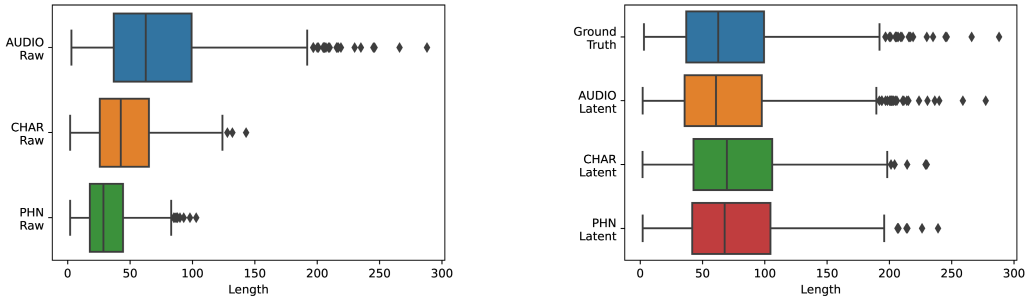
[9] Using SentencePiece: https://github.com/google/sentencepiece



(a) Grapheme-to-Phoneme

(b) Phoneme-to-Grapheme

Fig. 4. MAdmixture Aligner Ablation - Effects on Training



(a) Raw Encoder Output Lengths

(b) Aligned Latent Representation Lengths

Fig. 5. MAdmixture Aligner Evaluation - Length Distributions

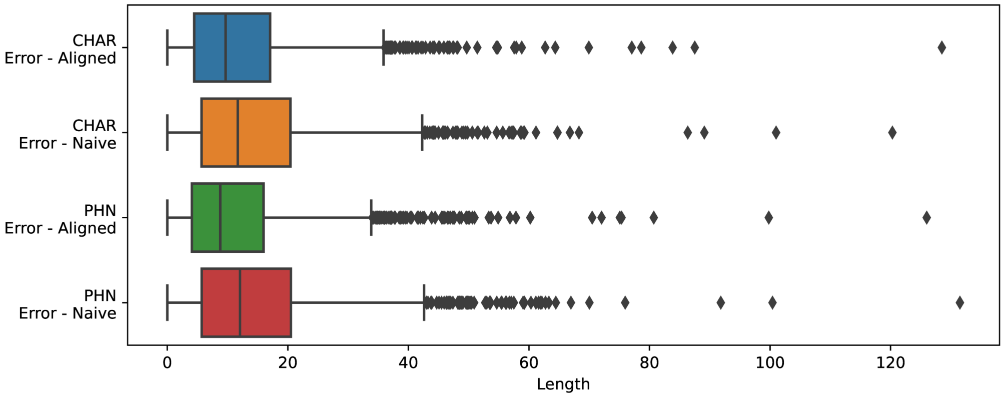


Fig. 6. MAdmixture Aligner Evaluation - Error Distribution

8.2. Baseline Model Architecture Details

Table 10 outlines the architecture that was used for the Audio/Grapheme/Phoneme MAdmixture model. c refers to the number of channels, k is the kernel size, d is the layer dimension, $p(drop)$ is the dropout probability, p is the pooling size, s is the scale parameter, ϵ is the spread parameter.

	mean	std	min	25%	50%	75%	max
AUDIO Raw Length	71.75	44.19	2.99	36.89	62.59	99.33	288.00
CHAR Raw Length	47.09	26.74	1.98	25.73	42.59	65.06	143.00
PHN Raw Length	32.23	18.34	1.97	17.76	28.65	44.36	103.00
AUDIO Latent Length	70.17	43.48	1.93	35.65	60.78	97.56	277.30
CHAR Latent Length	77.26	43.16	1.88	42.74	69.57	105.91	229.94
PHN Latent Length	75.99	42.81	1.97	41.72	67.71	104.56	239.02
AUDIO Error	1.57	0.94	0.05	0.95	1.37	1.96	10.70
AUDIO Naive Error	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CHAR Error	12.09	10.43	0.00	4.47	9.69	17.06	128.50
CHAR Naive Error	14.42	11.70	0.01	5.68	11.71	20.45	120.29
PHN Error	11.47	10.36	0.01	4.08	8.80	15.98	125.99
PHN Naive Error	14.59	11.89	0.00	5.72	12.07	20.51	131.49

Table 9. Lengths and Length Predictions - Descriptive Statistics

Modality	Encoder	Decoder	Aligner
Audio	**CRDNN** 2 × CNN $c = 128, 256, k = 3, p = 3$ 2 × Bi-LSTM, $d = 1024$ 2 × FC $d = 512$	**Tacotron 2** Prenet: $d = 128$ Attention: $d = 128, f = 32, k = 31$ RNN: $d = 1024$	**GMTW** $s = 1.0, \epsilon = 1.0$
Character	**GRU** 4 × GRU, $d = 512$ $p(\text{drop}) = 0.1$	**GRU** 4 × GRU, $d = 1024$ Attention: $d = 1024, k = 100$ $p(\text{drop}) = 0.15$	**GMTW** $s = 1.7, \epsilon = 1.0$
Phoneme	**GRU** 4 × GRU, $d = 512$ $p(\text{drop}) = 0.1$	**GRU** 4 × GRU, $d = 1024$ Attention: $d = 1024, k = 100$ $p(\text{drop}) = 0.15$	**GMTW** $s = 2.5\epsilon = 1.0$

Table 10. MAdmixture Baseline Architecture Hyperparameters

8.3. Curriculum Learning

Most speech processing systems involve training a model on sequences with long-term dependencies. Based on previous experience with training G2P systems, such as SoundChoice G2P [**28**], we hypothesize that model training can benefit from pre-conditioning on shorter sequences. The approach introduces a curriculum learning sampler that makes it possible to train the system with samples of increasing length using LibriSpeech [**25**] with LibriSpeech-Alignments [**22**] or any other compatible dataset with phoneme annotations. For a given epoch, the maximum and minimum sample length, in words, is given as a hyperparameter. First, the algorithm samples a length from a uniform distribution $l_{\text{sample}} \sim \text{Uniform}(l_{\text{min}}, l_{\text{max}})$ for each sample. Secondly, it samples an offset fraction from a uniform distribution $o_{\text{rel}} \sim \text{Uniform}(0, 1)$, which is then interpreted as a relative offset, i.e. a o_{rel} corresponds to selecting the first l_{sample} words, and $o_{\text{rel}} = 1.0$ corresponds to selecting the last l_{sample} words. The method can be used both within a single training run, gradually increasing sequence length at every n epochs (where n is a hyperparameter), and gradually over time as the model evolves. Table 11 shows the proposed curriculum for training MAdmixture.

Curriculum Step	Epoch	Min Words	Max Words	Sample Count
1	1	1	3	25000
2	20	2	10	50000
3	40	2	20	100000

Table 11. MAdmixture Learning Curriculum

We conduct experiments involving only this sampling curriculum and a variation where complete random samples of complete utterances are used. We find that the curriculum facilitates convergence, and a switch to full sequences results in better performance.

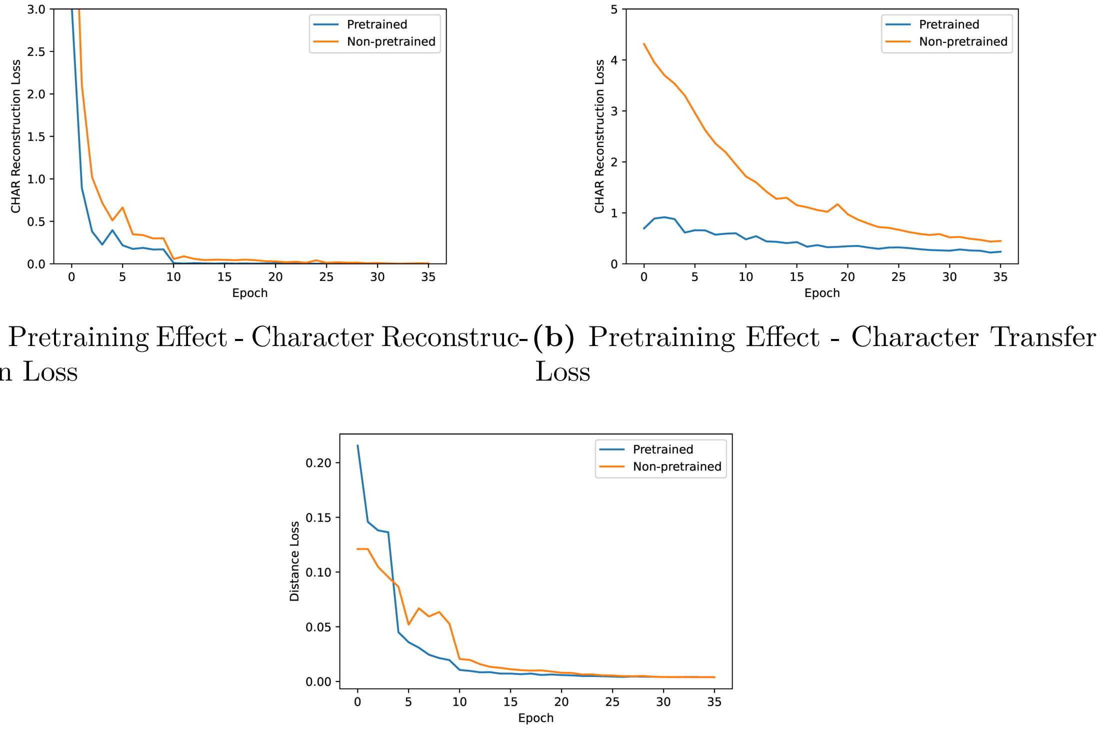
We use an initial learning rate of 0.001, annealed using the NewBob [10] scheduler with an annealing factor of 0.9 and an improvement threshold of 0.0025.

We find that the best results are achieved with very aggressive gradient clipping, with the maximum gradient set to 0.0001, with larger values frequently causing training instability. This could be due to the fact that a very large number of sub-networks is being trained jointly, and a large gradient update destroying learned representations can affect multiple sub-networks simultaneously.

8.4. Transfer Learning

The MAdmixture approach allows for the incorporation of existing pretrained models trained for a specific modality transfer, such as ASR or TTS and expand it to handle additional transfer directions and autoencoding. The proposed approach is to introduce an alternate *invertible projection aligner* that always produces an identity alignment matrix but uses a projection layer (1D convolutions) followed by an optional invertible global normalization layer that makes it possible to project pretrained representations to a common latent space and recover them before passing them to the decoder. This is combined with a modified training curriculum where initially both the parameters of source modality's encoder and the target modality's decoder are frozen, and only the aligner is trained together with the source modality's decoder, the target modality's encoder and any other modalities. After it is trained sufficiently to avoid destroying the pretraining, we then unfreeze the components, allowing the model full latitude to realign. Figure 7 shows the evolution of the character reconstruction loss, audio-to-character transfer loss and difference loss. We fined that starting with a pre-trained model has a beneficial effect on both the speed of learning and the speed with which audio and character latent spaces converge to a shared space. No noticeable effect

[10] Herve A Bourlard and Nelson Morgan, Connectionist speech recognition: a hybrid approach, vol. 247, Springer Science & Business Media, 2012.



(a) Pretraining Effect - Character Reconstruction Loss

(b) Pretraining Effect - Character Transfer Loss

(c) Pretraining Effect - Distance Loss

Fig. 7. MAdmixture Pretraining Effects

is observed on the speed of learning TTS if the audio encoder and character decoder are pretrained.

8.5. Hyperparameters and Hyperparameter Optimization

For token sequence modalities (characters and phonemes), we use a standard beam search implementation with a language model for the same token sequences also pre-trained on LibriSpeech data to help disambiguate sequences using linguistic regularities. We also use Oríon[11] for hyperparameter tuning with the Character Error Rate as the objective.

Table 12 shows the beam search hyperparameters used to achieve the listed results, following the Oríon-based hyperparameter search.

Given the large size of the search space and the time it takes to obtain results, we do not perform a hyperparameter search on the architecture. Instead, we use hyperparameters from existing SpeechBrain [**29**] ASR and TTS models with minor manual modifications.

[11] Xavier Bouthillier and Christos Tsirigotis and François Corneau-Tremblay and Thomas Schweizer and Lin Dong and Pierre Delaunay and Fabrice Normandin and Mirko Bronzi and Dendi Suhubdy and Reyhane Askari and Michael Noukhovitch and Chao Xue and Satya Ortiz-Gagné and Olivier Breuleux and Arnaud Bergeron and Olexa Bilaniuk and Steven Bocco and Hadrien Bertrand and Guillaume Alain and Dmitriy Serdyuk and Peter Henderson and Pascal Lamblin and Christopher Beckham, Epistimio/orion: Asynchronous Distributed Hyperparameter Optimization https://doi.org/10.5281/zenodo.3478592

Modality	Data	Hyperparameter	Value
CHAR	Segment	Minimum Decode Ratio	0.0001883
		Maximum Decode Ratio	1.206
		Beam Size	29
		EOS Threshold	4.202
		Maximum Attention Shift	235
		Language Model Weight	0.3729
CHAR	Full	Minimum Decode Ratio	0.00927
		Maximum Decode Ratio	1.931
		Beam Size	6
		EOS Threshold	1.057
		Maximum Attention Shift	125
		Language Model Weight	0.01981
PHN	Segment	Minimum Decode Ratio	0.002961
		Maximum Decode Ratio	1.231
		Beam Size	27
		EOS Threshold	3.625
		Maximum Attention Shift	125
		Language Model Weight	0.3495

Table 12. MAdmixture Beam Search Hyperparameters

For audio representations we re-use the hyperparameters used in the SpeechBrain [**29**] LibriSpeech [**25**] ASR and Tacotron [**35**] implementations, respectively.

8.6. Sampling

In a preliminary generation experiment, we train a UNet [**31**] with layers of 64, 128 and 256 channels, group normalization with groups of 32 neurons, 2 residual blocks per layer and a single attention layer. We find that decoding the generated latent space with multiple modalities results in close-sounding, albeit not identical decodings.

Example:

(1) CHAR: INTO SO MOON BEER
PHN: IH-N-T-AH-S-AH-N-D-AH

(2) CHAR: WHAND THEN AND SEES EARED SEE
PHN: HH-AE-N-D-DH-EH-N-EY-N-T-S

(3) CHAR: DEVICA A HEAP THE
PHN: D-EH-V-OY-AH-N-T-HH-IH-M

Further work may explore training the diffusion model non-jointly against the encoders of a model trained to convergence with join training and using additional techniques to fully align generated decodings.

Kind	Hyperparameter	Value
Input	Spec Module	Filterbanks
	Number of FFTs	400
	Number of MELs	40
	Minimum Frequency	0 Hz
	Maximum Frequency	8000 Hz
	Power	2
	Reference	1.0
Output	Spec Module	MEL Spectrogram
	Number of FFTs	1024
	Number of MELs	80
	Minimum Frequency	0 Hz
	Maximum Frequence	80 Hz
	Hop Length	256
	Window Length	1024
	Norm	Slaney
	Power	1
	Reference	10

Table 13. Spectrogram Hyperparameters

8.7. Alternative Transformer-Based Model

We attempt to build a preliminary proof-of-concept Transformer-based [**41**] model is similar to the CRDNN model in its basic structure; however, the individual encoders and decoders for token sequence modalities (characters and phonemes) are based on the Transformer [**41**] and Conformer [**13**] implementations of SpeechBrain [**29**]'s ASR models, whereas the TTS decoder is based on the TransformerTTS [**21**] architecture from Microsoft Research with some implementation and enhancement ideas from existing open-source implementations [**7**, **34**]. The G2P model achieves a token error rate of 3.2% for character-to-character and 1.4% for character-to-phoneme conversion on LibriSpeech samples of up to 20 words with a beam size of 1, which is close to the results RNN model - and these preliminary results were obtained without beam tuning. Preliminary ASR TER was 5.2% (also with beam size = 1), indicating that more training and tuning is needed. The hyperparameters are similar to those of SpeechBrain [**29**] ASR models: 8-layer encoders, 6-layer decoders, model dimension = 512, FFN dimension = 2048. Subsequent work on Transofrmers will involve tuning the TTS decoder and beam search, as well as incorporating a language model.

Figure 8 shows a training trajectory comparison between the CRDNN model and the Conformer model, as measured on the validation set with spectrogram augmentation enabled.

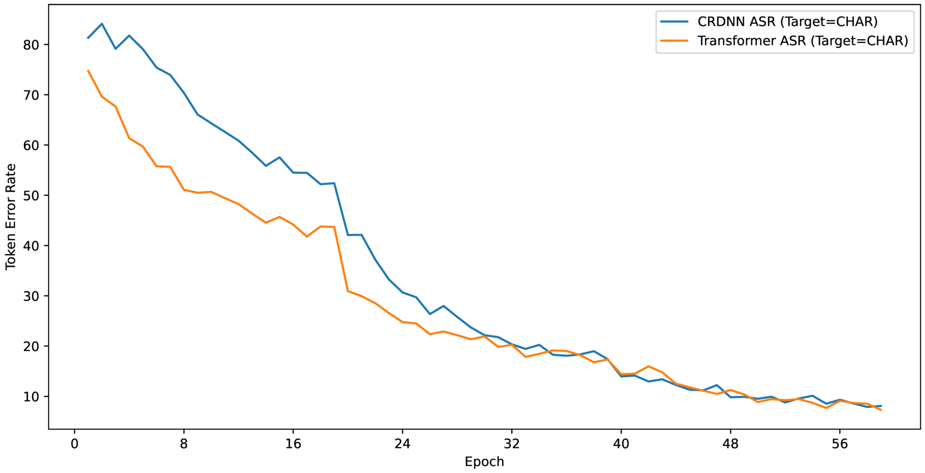


Fig. 8. MAdmixture: Conformer vs CRDNN Training Comparison

The trajectories are similar with the Conformer training slightly faster in the beginning. However, the last (full-sequence) curriculum step exhibited training instabilities that are likely due to the model overfitting to the sequence lengths on which it was trained. Subsequent work will address them with regularization and position sampling techniques. Overall, the preliminary experiment suggests that Transformer/Conformer [**13**, **41**] encoders and decoders are a promising direction for MAdmixture given that their fully-tuned single-task equivalents outperform their CRDNN counterparts.

References

[1] Bhuvan Agrawal, Markus Müller, Martin Radfar, Samridhi Choudhary, Athanasios Mouchtaris, and Siegfried Kunzmann. Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding, 2020.

[2] Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing, 2021.

[3] Zhongxin Bai and Xiao-Lei Zhang. Speaker recognition based on deep learning: An overview, 2021.

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM, 2009.

[5] Andreas Blattmann, Robin Rombach, Kaan Oktay, and Björn Ommer. Retrieval-augmented diffusion models, 2022.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[7] Axel Springer News Media Tech GmbH Co. KG Ideas Engineering. Transformertts. `https://github.com/as-ideas/TransformerTTS`, 2023.

[8] Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125, 2020.

[9] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. *CoRR*, abs/2007.01852, 2020.

[10] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all, 2023.

[11] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA, 2006. Association for Computing Machinery.

[12] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1764–1772, Bejing, China, 22–24 Jun 2014. PMLR.

[13] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition, 2020.

[14] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically, 2017.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[16] Ye Jia, Michelle Tadmor Ramanovich, Tal Remez, and Roi Pomerantz. Translatotron 2: High-quality direct speech-to-speech translation with voice preservation, 2022.

[17] Ieva Kazlauskaite, Carl Henrik Ek, and Neill Campbell. Gaussian process latent variable alignment learning. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 748–757. PMLR, 16–18 Apr 2019.

[18] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of EMNLP*, 2018.

[19] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February 1966.

[20] Jinyu Li. Recent advances in end-to-end automatic speech recognition, 2022.

[21] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu, and Ming Zhou. Neural speech synthesis with transformer network, 2018.

[22] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. Speech Model Pre-Training for End-to-End Spoken Language Understanding. In *Proc. Interspeech*, 2019.

[23] Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. Dialoglue: A natural language understanding benchmark for task-oriented dialogue, 2020.

[24] OpenAI. Guided diffusion. `https://github.com/openai/guided-diffusion/`, 2023.

[25] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *Proc. of ICASSP*, 2015.

[26] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*. ISCA, sep 2019.

[27] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning, 2017.

[28] Artem Ploujnikov and Mirco Ravanelli. SoundChoice: Grapheme-to-Phoneme Models with Semantic Disambiguation. In *Proc. Interspeech 2022*, pages 486–490, 2022.

[29] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. Speechbrain: A general-purpose speech toolkit, 2021.

[30] Markéta Rezácková, Jan Svec, and Daniel Tihelka. T5g2p: Using text-to-text transfer transformer for grapheme-to-phoneme conversion. In *Interspeech*, 2021.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[32] Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation, 2023.

[33] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[34] Soobin Seo. Transformer-tts. `https://github.com/soobinseo/Transformer-TTS`, 2023.

[35] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. *CoRR*, abs/1712.05884, 2017.

[36] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention is all you need in speech separation, 2021.

[37] Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, apr 2018.

[38] Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A survey on neural speech synthesis, 2021.

[39] Yun Tang, Hongyu Gong, Ning Dong, Changhan Wang, Wei-Ning Hsu, Jiatao Gu, Alexei Baevski, Xian Li, Abdelrahman Mohamed, Michael Auli, and Juan Pino. Unified speech-text pre-training for speech translation and recognition, 2022.

[40] Carnegie Mellon University. CMU Pronouncing Dictionary, July 2021.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[42] Phil Wang. denoising-diffusion-pytorch. `https://github.com/lucidrains/denoising-diffusion-pytorch`, 2023.

[43] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis, 2017.

[44] Xuesong Yang, Yun-Nung Chen, Dilek Hakkani-Tur, Paul Crook, Xiujun Li, Jianfeng Gao, and Li Deng. End-to-end joint learning of natural language understanding and dialogue manager, 2017.

Conclusion

This work explores a variety of deep learning approaches to the speech domain, including grapheme-to-phoneme conversion, speech recognition and speech synthesis in an attempt to build towards a more generic approach based on unified cross-modality signal representations, building on prior work in the domain. The SoundChoice [**28**] grapheme-to-phoneme system attempts to take advantage of mixed representations combining character tokens and sentence context in order to achieve significant performance improvements over baseline approaches by incorporating sentence context from pretrained models and by applying a variety of techniques from the speech recognition field that facilitate signal decoding. MAdmixture further expands on these ideas not only by using a similar decoding approach for text modalities but also by building on the Grapheme-to-Phoneme and combining it with elements of existing ASR and TTS models available within SpeechBrain [**29**] and introducing a new time-warping technique to create a single composite model incorporating all of these tasks in a single modular network - while also providing functionality similar to that of a classic autoencoder and a proof of concept for using common representations as a basis for a generative model. Empirical evaluation confirms the general feasibility of the approach. While the proof-of-concept MAdmixture models evaluated in the present work do not achieve state-of-the-art performance, further improvements can be attempted via architecture search, data transformation, the use of larger pre-trained language models and decoder fine-tuning for specific modalities.

Future work can build on the approaches explored and evaluated here to fine-tune the model to achieve levels of performance close to the state of the art on the relevant tasks while using aligned representations across modalities to which new modalities can be easily added. Some possible examples include lip-reading video, handwriting or token sequences for natural language understanding. It can also explore improved techniques of cross-modality alignment where the relative lengths of raw encoded samples vary more widely across modalities.

Within the broader context of artificial intelligence, the integration of information coming from different senses and different sources to produce a rich, vivid experience of the world as a coherent whole (rather than attending to only one type of signal at a time for a given task) and seamlessly transferring them across modalities is becoming key. As humans, we absorb it all, and then, when we read a book (text modality), and we say it out loud in our heads (speech), and our imagination completes a visual scene with sights, sounds, smells and feelings, allowing us to conjure up entire worlds, whether real or imaginary. State-of-the-art artificial intelligence with multimodality is already making strides towards our machines enhancing our experiences or creating their own - or, perhaps, even having their own if that is at all possible. And integrating senses or modalities is just a small building block of human intelligence, which then proceeds to correlate the fully integrated sensory information from the present moment with the context of time and space, broader history, histories that could have been, possible futures that may or may not arrive... It miraculously navigates this infinite realm of possibilities by making decisions and then taking action, little by little taking us towards better outcomes and greater possibilities, and in today's complex world, it is already difficult to imagine this navigation without the help of intelligent machines. This work on creating a cross-modal representational "language" in a very specific context is but a small learning experience, a tiny drop in the vast ocean of explorations in this field. I hope that with all the work in machine learning and the scientific community at large on multimodality we will come one step closer to understanding our own integrating experience, giving a small bit of it to our machines and then building a better world with their help.