Deep Learning for Communication

Emergence, Recognition and Synthesis

Ioannis Alexandros Assael

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2019

This thesis is dedicated to
those who light up my life.

Abstract

Human intelligence is a social phenomenon tightly coupled to the act and process of communication. Ever since the early prehistoric period, humans have been able to communicate amongst themselves at an unprecedented and unparalleled level compared to all other living species. Communication led humans to develop media such as the spoken and written word to effectively convey the meanings of concrete and abstract concepts, and still today a substantial part of human life is spent communicating and sharing information. The scientific study of communication began in Classical Greece with the work of Aristotle and was to evolve through time into the work on information theory by Claude E. Shannon. This work proposes three novel methods for studying the processes of emergence, recognition, synthesis and enhancement of communication, using recent advances in deep learning. The first method investigates the emergence of communication among agents, and introduces a differentiable way of learning communication protocols. The second studies speech recognition in visual verbal communication, and for the first time solves sentence-level lipreading with deep neural networks trained end-to-end. The third and final method proposes a meta-learning approach for sample efficient verbal communication via text-to-speech synthesis. This thesis advances deep learning in these areas, and defines the premises for the creation of novel technologies for the greater good of society.

Acknowledgements

I feel incredibly fortunate, privileged, and thankful for all the opportunities I have been given that led me to pursue this research thesis. It has been the most creative and exciting journey I could have ever imagined, and thanks to all the people that provided me with the foundations, knowledge, equipment, environment, financial well-being, support, courage and happiness to produce this work.

This thesis is only made possible with the close collaboration with my advisors Prof. Nando de Freitas and Prof. Shimon Whiteson. Prof. de Freitas' and Prof. Whiteson's way of thinking has profoundly shaped me to the person I am today, how I approach research problems, and most of all how I approach life. I'm incredibly thankful for their continuous support, inspiring counsel and guidance during the course of this work. I would also like to sincerely thank them for showing me, and teaching me what academic ethics, and excellency should be.

This thesis was also made possible by the great research environment that University of Oxford has provided me during my studies here. University of Oxford is a unique, collaborative, thriving environment that combines great history and tradition with the innovative and forward-thinking. This environment provided me the capabilities to pursue my passions throughout my time here, and make friendships that will last forever. This work was made possible by the support of the Oxford-Google DeepMind Graduate Scholarship, the EPSRC, and the generous hardware donations by NVIDIA. Thanks also to all of the administrative staff that have kept everything at the Department of Computer Science running smoothly, including Ms. Julie Sheppard, who always brought the happiest news with the kindest smile, and all the wonderful conversations we've shared. I am extremely grateful for finding such bright minds and friends in our lab as Dr. Jakob Foerster, who made every day in the lab to be remembered; and sharing my passions and goals with Mr. Brendan Shillingford, who would always share his knowledge and expertise, with clarity of thinking and a lot of fun and joy, and made this journey a story to be remembered. I am also thankful for all interactions, advice and guidance from my examiners Dr. Tim Rocktäschel and Prof. Phil Blunsom, Prof. Andrew Zisserman, Dr. Andrew Senior, Prof. Chris Dyer, Dr. Varun Kanade, Dr. Misha Denil, Dr. Matt Hoffman, Dr. Bobak Shahriari, Dr. Yutian Chen, Mr. Archit Gupta, Dr. Tom Walters, Dr. Caglar Gulcehre, Dr. Tom Paine, Dr. Cían

Hughes, Dr. Koray Kavukcuoglu, to the support of Ms. Lorrayne Bennett, Ms. Sarah Henderson, and Ms. Helen King, and to all of my close collaborators who contributed to projects of this thesis.

On a personal side, the environment of Oxford is also made unique because of the college life, the mixture of different cultures, and the interdisciplinary interactions. I would like to thank Thea Sommerschield, Lea Sefer, Jenny Vafiadou, Marco Molteni, Mitko Sabev, Miguel Blázquez Carretero, Juan José Riva Grela, Lanbo Zhang, (and the Wolfson College bar usual suspects), Vasilis Papadogiannis, Despoina Charou, and Ioanna Rota, for all the happy moments, the conversations we've shared, and for making Wolfson College one of the most special places I've ever lived. I would also like to thank Konstantinos Adraktas for all the moments and activities we've shared these years; and Fotis Touparis, Menelaos Tsitonas, Vaggelis Margaritis, Kostas Dimitropoulos, Panos Pantides, Odysseas Votsis, Orestis Milios and Kostas Papadopoulos, for showing me the power of friendships even a thousand miles away.

Finally, I would like to thank my parents Dora and Marc for raising me in a wonderful environment, for their constant faith, for encouraging me to pursue my interests, and for always being there for me, as well as, my grandmother Kitsa for her constant support.

Contents

1 Introduction

It seems relevant to begin a study on the interplay between communication and deep learning by noting the etymology of the word "communication", as it illustrates the scope, potential, and relevance of this thesis. "Communication" stems from the Latin - commūnis meaning "common, universal, public". The resonance of this stem is illustrated by the several disciplines which engage in the scientific study of communication, while its relevance is embodied in the coupling of the act of communication with human evolution, and technological progress. This chapter focuses on both these aspects, by examining how certain models of the communication process can be effectively adopted as guiding frameworks for new work in the field of deep learning. We begin with an overview of the main hypotheses concerning the very origins of communication, and the current established models to study and analyse it. We then employ one of the most broadly used models of the communication process as the guiding framework of this thesis, and proceed to study the emergence, recognition, and synthesis of communication across a series of selected case studies in the field of deep learning.

1.1 Communication

We experience and interact with our surroundings through our senses: in essence, this is a process of information gathering. The very same process occurs in most of the animal kingdom. For instance, the sound of running water can be used to infer the existence of a stream not far away. Likewise, the rattle of a rattlesnake's tail is also a powerful warning signal to other animals to keep their distance. This is a case of aposematism, where animals and plants warn predators by their appearance that they are toxic, foul, or dangerous. The ability to extract and process information from the surrounding environment is a measure of a species' intelligence [15], and is indeed crucial for its survival. However, in the aforementioned case of aposematism, it is clear that animals do not only process information, but they are also capable of generating it. Sounds and noises such as barks, roars, and even body gestures are all meant to be interpreted by other animals. According to Taylor [16], the process of deliberately generating information with the intent of it being interpreted by others is called communication. Within the entire animal kingdom, it was *Homo sapiens* whom since the early Prehistoric era was able to communicate at an unprecedented and unparalleled level compared to any other species [17]. Indeed, the act of communication has been tightly coupled to the evolution of humanity, to its intelligence, and its technological progress [18].

The narration of myths, folk tales, parables, and stories within a social setting ultimately allowed *Homo sapiens* to collaborate in large groups and in extremely flexible ways [19]. This ability has been described as a cognitive revolution for *Homo sapiens* because it implies the creation of an imagined space and time, distinct from physical reality and predicated on the exchange of stories and abstract concepts. The communication of an imagined reality set *Homo sapiens* apart from all other animals, and maximised this particular species' potential for large-scale collaboration and interaction by virtue of shared myths, belief systems, and histories. Within the human world, communication developed through mutually understood media such as speech and writing systems (among others), the latter eventually leading to

the advance of human prehistory into history, the period when humans began to resort to writing to convey more complex meanings in a durable manner. In today's world of technological revolution-cum-evolution, communication is still undergoing major changes and developments, not only regarding the means by which people communicate amongst themselves (e.g. radio, television, smartphones, satellites), but also the ways in which concepts borrowed from human communication systems are embedded in the very architecture of disciplines such as computer science (e.g. computer networks, encoder / decoder architectures, modulators, neural networks - to mention but a few examples of computer science's adaptation of models, terms and concepts inherent to human communication to its own tasks and frameworks).

Communication as a scientific field of study has existed since antiquity, and in the course of the twentieth century was split into the following fields [20]: communication studies, which focus on human communication and investigate communication through signs between living organisms and information theory, which studies the quantification, storage, and communication of information. The first steps of information theory were proposed by Claude E. Shannon in 1948 in his attempt to identify the limits of signal processing and communication operations such as data compression, in his landmark work "A Mathematical Theory of Communication" [21]. The model introduced by his work to explain the process of information transmission consists one of the guiding frameworks of this thesis.

The present thesis sets out to tackle some of the crucial steps in the communication process, namely emergence, recognition, and synthesis of communication (cf. following sections), using deep learning. Deep learning is a field of machine learning that allows statistical models to learn representations of data with multiple levels of abstraction. Deep learning models are inspired by biological neural networks, and the abstraction layers facilitate the discovery of intricate structures embedded within large amounts data. In tandem with the rapid expansion of computational power of recent years, these models have provided solutions to elusive challenges, from drug discovery by predicting how proteins fold [22] to learning computer programs [23–25], unsupervised machine translation [26] and reading comprehension [27], speech

recognition [28], text-to-speech synthesis [29], image-to-image translation [30], and notoriously difficult games such as Go [31] and StarCraft [32] (among others).

Compared to traditional machine learning methods, the success of neural network models lies in their ability to learn to represent high-dimensional information in a "compressed" manner. More specifically, each abstraction layer of a neural network can be though as a low-dimensional representation of the previous layer, where each representation is meaningful to the task being solved. Such low-dimensional representations could be also obtained using traditional feature extraction algorithms. However, the last few years the literature has shown that hard-coded feature extractors are limited and they are not necessarily optimal for any task being solved. Thus, the ability of neural networks to learn to reduce the dimensionality of high-dimensional data to low-dimensional meaningful vector representations has empowered the community to tackle complex problems that were not computationally tractable with traditional methods. Audio, images, videos, or the states of a simulated environment are some examples of high-dimensional data. These examples are also parts of different subprocesses of communication, and the capability to handle such complex inputs motivates the choice of deep learning in the present work. For further details on deep learning we refer the reader to the work of Goodfellow et al. [33], and the surveys of LeCun et al. [34] and Schmidhuber [35].

In the following two sections, we survey the basic types of communication and the main models explaining communication as a process. This is not an attempt to trace an in-depth analysis of recent discourse on communication theory. Instead, a concise overview of such expansive topics enables us to isolate certain key concepts inherent to communication theory, which can be effectively used as the guiding framework for this thesis' discussion of select advances in the field of deep learning. The methodological aim of this thesis consists in allowing innovative perspectives to arise from the interdisciplinary coupling of communication theory and deep learning. The outcome of this approach is intended as a collection of state-of-the-art contributions to technological advances within the scope of communication, aiding humans in different aspects of the communication process by overcoming

obstacles of a physical, medical or conceptual nature, and ultimately enhancing the communal aspect of communication.

1.1.1 Types of communication

The process of communication at its most essential consists in a server and a receiver which transfer information through a channel [21]. The channel may accept verbal, non-verbal, or other types communication [36].

Verbal communication

We will use the term "verbal" to refer the spoken and written conveyance of a message by means of sets of symbols. In large part of the literature, verbal communication is synonymous to "language" [16]. However, the definition of language as a method of human communication is beyond the scope of the present work. Verbal communication relies on a symbolic system consisting of a discrete set of words. The ways in which a relation is struck between these words and objects or meanings has been explained by Ogden and Richards [37]. In accordance with a set of rules (a grammar), these symbols can be manipulated and combined in sequences to form sentences.

Non-verbal communication

Non-verbal communication refers to the part of human communication that is not conveyed through symbols, but through visual, auditory, tactile, and kinesthetic (physical) channels instead. A substantial portion of human communication is non-verbal, and it includes visual cues such as body language or facial expressions, factors such as distance and physical environments, variables such as variations and qualities of voice (e.g. rate, pitch, prosody and loudness) and of touch (haptics). With very few exceptions, these codes provide information and messages related to the current conditions, place or time [38].

It must be noted, as it will be of interest later in this thesis, that contrary to popular belief, visual methods of communication such as sign language and lipreading are considered verbal communication. This is because the vocabulary, grammar, and other linguistic structures of such visual communication methods follow similar or identical classifications as those of spoken or written languages. However, because of qualities such as the intensity, the speed, and the size of mouth movements or signs, these visual methods of communication are also understood as conveyors of non-verbal cues of communication.

1.1.2 Models of communication

Communication theory makes use of conceptual models to explain the process of communication. For the most part, these models follow the basic concept (outlined above) of sending and receiving messages through a channel [39], and are used to illustrate the interactions between the participating elements in the communication act. Several different attempts have been made to explain the communication process in different settings and through different perspectives. In this section, we will analyse some of the most commonly used ones, which are also closely related to the models discussed in the deep learning case studies appearing in the following chapters.

Aristotle's model (4^{th} century BC)

One of the earliest models of communication was given by the ancient Greek philosopher Aristotle (384 - 322 BCE). He is credited with developing the basic teachings of the art and discipline of rhetoric. The model is presented in his treatise *Rhetoric* (Ῥητορική) [40]. Written in the 4th century BC, it focuses on the subtle art of persuasion through the spoken work. As illustrated in Figure 1.1, Aristotle identifies five elements in communication [41]: the speaker, the speech, the occasion for the speech to take place, the audience, and the effect it causes. The model is a simple and linear one. Linear models are unidirectional and do not contain feedback

loops. This model is therefore targeting human communication, specifically public speaking, rather than interpersonal communication [42].

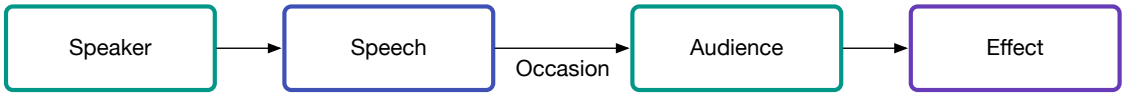


Figure 1.1: Aristotle's model [40], targeting human mass communication. It identifies five elements in communication, the speaker, the speech, the occasion that the speech takes place, the audience, and the effect it causes.

Lasswell's model (1948)

Another widely referenced early linear model is Harold D. Lasswell's. In his 1948 work [43], Lasswell introduced a model that strikes as a more generalised version of Aristotle's. The model suggests a message flow occurring within a pluralistic society where many possible channels exist [42]. As can be seen in Figure 1.2, the channel itself is an entity in the model.

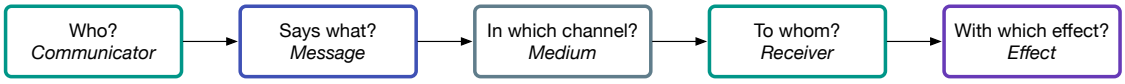


Figure 1.2: Lasswell's model of communication [43] suggests a message flow in a pluralistic society where there are many channels. It has many similarities with Aristotle's model, and is based on the following questions "who?", "says what?", "in which channel?", "to whom?" and "with which effect?".

Shannon and Weaver's model (1948, 1949)

In 1948, Claude E. Shannon presented one of the most influential works on communication theory in his work "A Mathematical Theory of Communication" [21], which was subsequently co-authored in a 1949 book with Warren Weaver "The Mathematical Theory of Communication" [44]. The Shannon-Weaver model treats communication as a transmission of messages. Their work was developed during the Second World War in the American Bell Telephone Laboratories, and studies

the efficiency of multiple communication channels. The available channels at the time were telephones and radio waves, and the goal was to optimise the capacity of both [38]. However, such channels had a limited capacity and additional sources of noise, making the information liable to corruption. To tackle this issue, Shannon developed the concept of information entropy as a measure for the uncertainty of a message's content. His invention marked the beginning of information theory.

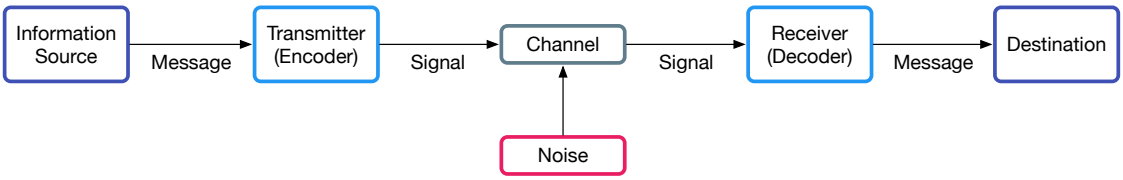


Figure 1.3: Shannon and Weaver's model of communication [21, 44].

Using the elements of the Shannon-Weaver model in Figure 1.3, we can break down the process of communication into the following six basic elements, which also recur in the majority of subsequent literature on the topic [39].

- **Source**: also referred to "information source", which produces a message or sequence of messages to be communicated to the receiving terminal.
- **Transmitter**: also called as "sender" or "speaker" [41], which encodes the message in a signal.
- **Channel**: the medium used to transmit the signal from sender to receiver. Channels contain an amount of "noise" that can corrupt signals.
- **Receiver**: which reconstructs the message from the signal received, carrying out the sender's process in reverse.
- **Destination**: which is the person (or object) for whom the message is intended.
- **Message**: a concept, information, or statement that is sent in a verbal or non-verbal form to the recipient.

The main criticism directed to the Shannon-Weaver model is that it depicts communication as a unidirectional process, whereas communication in its essence involves a bidirectional dynamic. Later works such as Schramm [45] and DeFleur [46] include this feedback loop. For further analysis we refer the reader to the works by Craig [39], Griffin [47], Narula [42] and Fiske [38].

Taylor's model (2009)

Intended as an expansion upon the Shannon-Weaver model, but focussing instead on the content transmitted at each step, Taylor [16] presents a broader model explaining the process of communication. His model is illustrated in Figure 1.4 and was used to explain technologies used for communication such as speech synthesis. In the model, the sender starts with a meaning intended for transfer and generates a message; the message is encoded in a signal and then transferred to the receiver. The receiver does the inverse process and decodes the signal to a message, which is then mapped to a meaning by the process, which ultimately consists in the act of understanding.

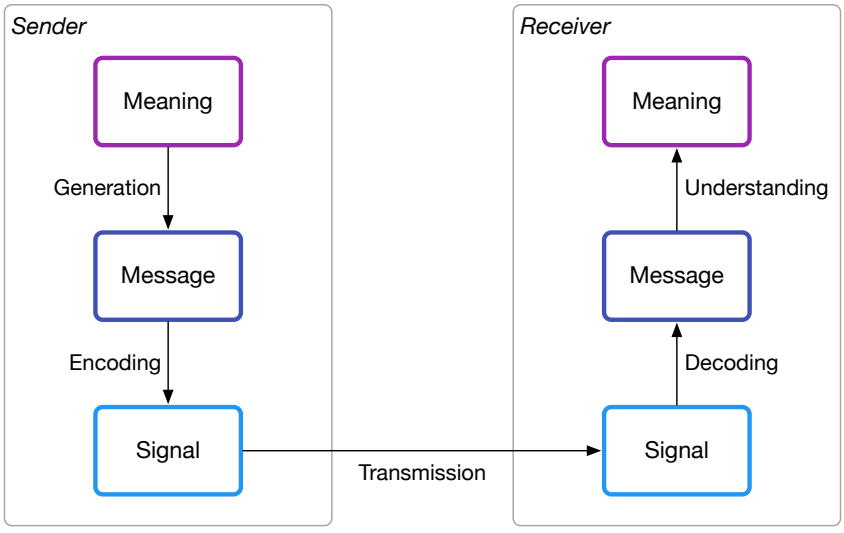


Figure 1.4: Taylor's model of communication [16].

1.2 Motivation & contributions

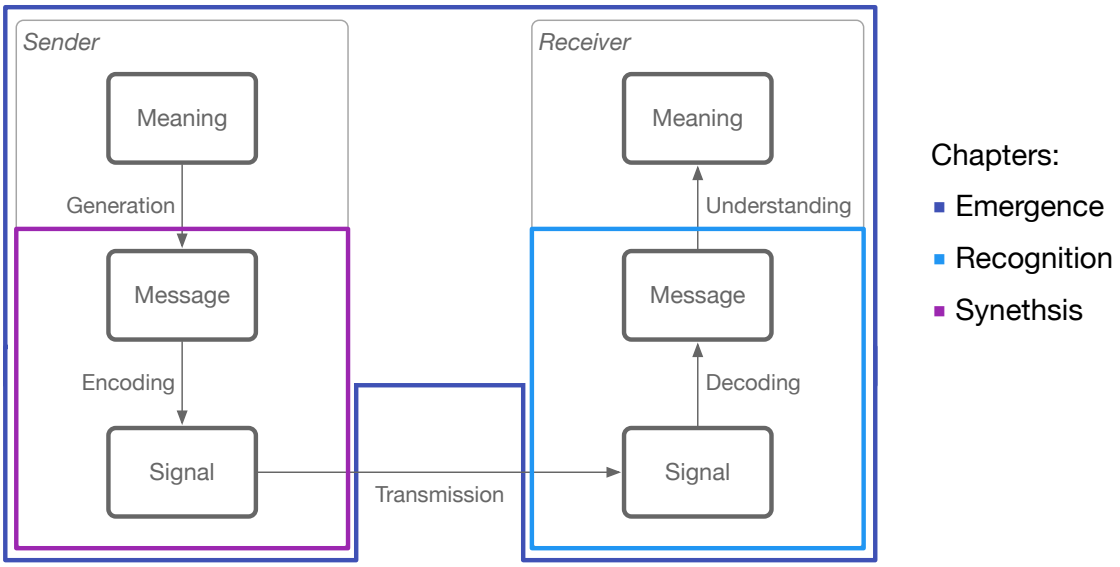


Figure 1.5: This figure illustrates an overview of the chapters of this thesis based on selected subprocesses of Taylor's model of communication.

Taylor's model consists in the guiding framework of the present work. Because this thesis addresses the process of communication from the perspective of machine learning and deep learning, it is the latter communication model [16] which effectively renders more and can effectively incorporate the technological advances of the 21st century. By using Taylor's model of communication as a guiding framework, we are able to isolate three discrete subprocesses. The subprocesses are outlined in Figure 1.5 and comprise the emergence, recognition, and synthesis of communication. Each of these subprocesses inspires and guides the content of each of the chapters in this thesis. In Chapter 2 we study the emergence of communication between intelligent agents and a limited capacity channel. Recognition is the act of decoding the signal to a message; Chapter 3 outlines recent advances in the automated recognition of verbal communication, such as lipreading. Finally, the reverse process of message encoding to a signal is called synthesis, and in Chapter 4 we present a novel method for speech synthesis drawing on a limited amount of data.

Taylor's model was adapted as a guiding framework because of it's flexibility to generalise and study a plethora of different communication problems and settings, from human-to-human to machine-to-machine information exchange. Each of these different settings affects directly all the subprocesses described in the model, and thus, the crucial component that is indirectly stated but missing as a separate entity in Taylor's model is the "environment". One can think of the environment, as the physical or simulated setting, that the sender and the receiver are part of and are governed by its laws or rules. A similar concept of the surroundings appears also in the study of Westley and MacLean Jr [48]. Such a set of surrounding rules directly affects the way communication can emerge, be generated and understood. Disentangling the environment from Taylor's model as a separate entity could bring additional clarity and allow a better study of different communication processes. For the scope of this thesis, we will assume that the sender, the receiver, and the transmission channel are all part of an environment which preside over the process of communication.

1.2.1 Emergence

The first chapter considers the problem of multiple agents sensing and acting in environments to maximise their shared utility. In these environments, agents must learn communication protocols to share information that is needed to solve the tasks. The chapter investigates the emergence of communication, provided a limited bandwidth channel similar to the one described by Shannon [21], how these protocols can be learnt using reinforcement learning, and whether learning would be more effective if done end-to-end. To our knowledge, this is the first work to demonstrate end-to-end learning of protocols in complex environments by embracing deep neural networks. The environments are co-operative general-sum games and inspired by communication riddles and multi-agent computer vision problems with partial observability. We propose two approaches for learning in these domains: Reinforced Inter-Agent Learning (RIAL) and Differentiable Inter-Agent

Learning (DIAL). Specifically, the former uses deep Q-learning, while the latter exploits the fact that, during learning, agents can backpropagate error derivatives through (noisy) communication channels. This approach therefore makes use of centralised learning, but decentralised execution. Our experiments introduce new environments for studying the learning of communication protocols, and present a set of engineering innovations that are essential for success in these domains.

The two works presented in this chapter, RIAL and DIAL, were joint first co-authorships with Dr. Jakob Foerster. My contributions consisted in the model architecture research, the stability of the proposed algorithms, and the experimental framework that was later open-sourced to aid further research in the field.

Related publications:

[1] J. N. Foerster*, Y. Assael*, N. de Freitas, and S. Whiteson. Learning to communicate to solve riddles with deep distributed recurrent Q-networks. In *International Joint Conference on Artificial Intelligence Workshop*, 2016.

[2] J. N. Foerster*, Y. Assael*, N. de Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.

1.2.2 Recognition

The second chapter focuses on methods for recognising visual verbal communication such as lipreading, also known as, visual speech recognition. Lipreading is the task of decoding text from the movement of a speaker's mouth, and has an enormous potential to medial applications. This chapter studies whether by using deep learning, sentence-level lipreading can be tackled more effectively, and how could it positively impact the experience of patients with speech impairments.

We present two state-of-the-art methods LipNet, and Vision-to-Phoneme (V2P). LipNet is the first model to solve end-to-end sentence-level lipreading, mapping videos of lips to sequences of character distributions. V2P is our follow-up work,

presenting an improved large-scale system consisting of a video processing pipeline that maps raw video to stable videos of lips and sequences of phonemes, a scalable deep neural network that maps the lip videos to sequences of phoneme distributions, and a production-level speech decoder that outputs sequences of words. Both works conduct comparisons with professional lipreaders to illustrate the potential impact of our work. In the case of, V2P we also construct the largest existing real-world lipreading dataset, consisting of pairs of transcriptions and video clips of faces speaking from YouTube. V2P represents a significant improvement from previous lipreading approaches, including variants of LipNet and of Watch, Attend, and Spell (WAS).

The two works presented in this chapter, LipNet and V2P, were joint first co-authorships with Mr. Brendan Shillingford. My contributions in LipNet comprised the efficient data processing pipeline, the model architecture research for solving lipreading, conducting the experimental and human evaluations, and working with expert linguists to analyse the saliency maps produced. In the case of V2P, I focused on model research and its scalability, from an architecture and engineering perspective. Furthermore, I wrote a complex parallel data processing pipeline to scale up the video processing to all of YouTube and introduced novel smoothing and filtering characteristics which improved our model's performance.

Related publications:

[3] Y. Assael*, B. Shillingford*, S. Whiteson, and N. de Freitas. LipNet: End-to-end sentence-level lipreading. In *GPU Technology Conference*, 2017.

[4] B. Shillingford*, Y. Assael*, M. W. Hoffman, T. Paine, C. Hughes, U. Prabhu, H. Liao, H. Sak, K. Rao, L. Bennett, M. Mulville, M. Denil, B. Coppin, B. Laurie, A. Senior, and N. de Freitas. Large-scale visual speech recognition. In *INTERSPEECH*, 2019.

1.2.3 Synthesis

The third chapter studies the synthesis of verbal communication from a given text. However, such methods require an enormous amount of speech data. This chapter investigates whether meta-learning can be used to improve the sample efficiency of these methods allowing them to be used in limited data medical settings. More specifically, it presents SEA a meta-learning approach for adaptive text-to-speech (TTS) with few data. During training, a multi-speaker model is learnt using a shared conditional WaveNet core and independently learned embeddings for each speaker. The aim of training is not to produce a neural network with fixed weights, to be then deployed as a TTS system. Instead, the aim is to produce a network that requires limited data at deployment time, so as to rapidly adapt to new speakers. Our experimental evaluation shows that these approaches are successful at adapting the multi-speaker neural network to new speakers, obtaining state-of-the-art results in both sample naturalness and voice similarity with merely a few minutes of audio data from new speakers.

My contributions in this work focused on the generation of speaker embeddings, prosody modelling using learnable speaker embeddings and the analysis of synthetic examples using a production level speaker identification model.

Related publications:

[5] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie, C. Gulcehre, A. van den Oord, O. Vinyals, and N. de Freitas. Sample efficient adaptive text-to-speech. In *International Conference on Learning Representations*, 2019.

1.3 Summary of publications

The following publications have stemmed from this work, and comprise the core of this thesis. The listed publications with a "*" indicate joint first co-authorships.

[1] J. N. Foerster*, Y. Assael*, N. de Freitas, and S. Whiteson. Learning to communicate to solve riddles with deep distributed recurrent Q-networks. In *International Joint Conference on Artificial Intelligence Workshop*, 2016.

[2] J. N. Foerster*, Y. Assael*, N. de Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.

[3] Y. Assael*, B. Shillingford*, S. Whiteson, and N. de Freitas. LipNet: End-to-end sentence-level lipreading. In *GPU Technology Conference*, 2017.

[4] B. Shillingford*, Y. Assael*, M. W. Hoffman, T. Paine, C. Hughes, U. Prabhu, H. Liao, H. Sak, K. Rao, L. Bennett, M. Mulville, M. Denil, B. Coppin, B. Laurie, A. Senior, and N. de Freitas. Large-scale visual speech recognition. In *INTERSPEECH*, 2019.

[5] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie, C. Gulcehre, A. van den Oord, O. Vinyals, and N. de Freitas. Sample efficient adaptive text-to-speech. In *International Conference on Learning Representations*, 2019.

Further published work during my doctoral thesis in the broader field of machine learning.

[6] M. J. Assael, K. D. Antoniadis, I. N. Metaxa, S. K. Mylona, Y. Assael, J. Wu, and M. Hu. A novel portable absolute transient hot-wire instrument for the measurement of the thermal conductivity of solids. *International Journal of Thermophysics*, 36(10-11):3083–3105, 2015.

[7] H. Mossalam, Y. Assael, D. M. Roijers, and S. Whiteson. Multi-objective deep reinforcement learning. In *Advances in Neural Information Processing Systems Deep Reinforcement Learning Workshop*, 2016.

[8] R. Ponte Costa*, Y. Assael*, B. Shillingford*, N. de Freitas, and T. Vogels. Cortical microcircuits as gated-recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 272–283, 2017.

[9] Y. Assael, K. D. Antoniadis, and M. J. Assael. From analog timers to the era of machine learning: The case of the transient hot-wire technique. In *Thermophysics*, volume 1866, page 020001. AIP Publishing, 2017.

[10] N. D. Goumagias, D. Hristu-Varsakelis, and Y. Assael. Using deep Q-learning to understand the tax evasion behavior of risk-averse firms. *Expert Systems with Applications*, 101:258–270, 2018.

[11] A. Gupta, B. Shillingford, Y. Assael, and T. C. Walters. Speech bandwidth extension with wavenet. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2019.

[12] B. Shillingford, Y. Assael, and M. Denil. Interactive decoding of words from visual speech recognition models. In *arXiv preprint*, 2019.

[13] T. Makino, H. Liao, Y. Assael, B. Shillingford, B. Garcia, O. Braga, and O. Siohan. Recurrent neural network transducer for audio-visual speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE, 2019.

[14] Y. Assael*, T. Sommerschield*, and J. Prag. Restoring ancient text using deep learning: a case study on Greek epigraphy. In *Empirical Methods in Natural Language Processing*, 2019.

2 Emergence

We consider the problem of multiple agents sensing and acting in environments to maximise their shared utility. In these environments, agents must learn communication protocols to share information that is needed to solve the tasks. This chapter investigates how these protocols can be learnt using reinforcement learning, and whether learning would be more effective if done end-to-end. The environments are inspired by communication riddles and multi-agent computer vision problems with partial observability. We propose two approaches for learning in these domains: Reinforced Inter-Agent Learning (RIAL) and Differentiable Inter-Agent Learning (DIAL). The former uses deep Q-learning, while the latter exploits the fact that, during learning, agents can backpropagate error derivatives through (noisy) communication channels. Hence, this approach uses centralised learning but decentralised execution. Our experiments introduce new environments for studying the learning of communication protocols and present a set of engineering innovations that are essential for success in these domains.

2.1 Introduction

How language and communication emerge among intelligent agents has long been a topic of intense debate. Among the many unresolved questions are: Why does language use discrete structures? What role does the environment play? What is innate and what is learned? And so on. Some of the debates on these questions have been so fiery that in 1866 the French Academy of Sciences banned publications about the origin of human language [49].

The rapid progress in recent years of machine learning, and deep learning in particular, opens the door to a new perspective on this debate. How can agents use machine learning to automatically discover the communication protocols they need to coordinate their behaviour? What, if anything, can deep learning offer to such agents? What insights can we glean from the success or failure of agents that learn to communicate?

In this work, we take the first steps towards answering these questions. Our approach is programmatic: first, we propose a set of multi-agent benchmark tasks that require communication; then, we formulate several learning algorithms for these tasks; finally, we analyse how these algorithms learn, or fail to learn, communication protocols for the agents. The tasks that we consider are fully cooperative, partially observable, sequential multi-agent decision making problems. All the agents share the goal of maximising the same discounted sum of rewards. While no agent can observe the underlying Markov state, each agent receives a private observation correlated with that state. In addition to taking actions that affect the environment, each agent can also communicate with its fellow agents via a discrete limited-bandwidth channel. Due to the partial observability and limited channel capacity, the agents must discover a communication protocol that enables them to coordinate their behaviour and solve the task.

We focus on settings with *centralised learning* but *decentralised execution*. In other words, communication between agents is not restricted during learning, which is performed by a centralised algorithm; however, during execution of the learned

policies, the agents can communicate only via the limited-bandwidth channel. While not all real-world problems can be solved in this way, a great many can, e.g., when training a group of robots on a simulator. Centralised planning and decentralised execution is also a standard paradigm for multi-agent planning [50, 51].

To address this setting, we formulate two approaches. The first, *reinforced inter-agent learning* (RIAL), uses deep Q-learning [52] with a recurrent network to address partial observability. In one variant of this approach, which we refer to as *independent Q-learning*, the agents each learn their own network parameters, treating the other agents as part of the environment. Another variant trains a single network whose parameters are shared among all agents. Execution remains decentralised, at which point they receive different observations leading to different behaviour. The second approach, *differentiable inter-agent learning* (DIAL), is based on the insight that centralised learning affords more opportunities to improve learning than just parameter sharing. In particular, while RIAL is end-to-end trainable *within* an agent, it is not end-to-end trainable *across* agents, i.e., no gradients are passed between agents. The second approach allows real-valued messages to pass between agents during centralised learning, thereby treating communication actions as bottleneck connections between agents. As a result, gradients can be pushed through the communication channel, yielding a system that is end-to-end trainable even across agents. During decentralised execution, real-valued messages are discretised and mapped to the discrete set of communication actions allowed by the task. Because DIAL passes gradients from agent to agent, it is an inherently deep learning approach.

Experiments on two benchmark tasks, based on the MNIST dataset and a well known riddle, show, not only can these methods solve these tasks, they often discover elegant communication protocols along the way. To our knowledge, this is the first time that either differentiable communication or reinforcement learning with deep neural networks has succeeded in learning communication protocols in complex environments involving sequences and raw images. The results also show that deep learning, by better exploiting the opportunities of centralised learning, is a uniquely

powerful tool for learning such protocols. Finally, this study advances several engineering innovations that are essential for learning communication protocols in our proposed benchmarks.

2.2 Multi-agent reinforcement learning

2.2.1 Related work

Research on communication spans many fields, e.g. linguistics, psychology, evolution and artificial intelligence. In artificial intelligence, it is split along a few axes: a) predefined or learned communication protocols, b) planning or learning methods, c) evolution or RL, and d) cooperative or competitive settings.

Given the topic of our work, we focus on related work that deals with the cooperative learning of communication protocols. Out of the plethora of work on multi-agent RL with communication, e.g., [53–56], only a few fall into this category. Most assume a pre-defined communication protocol, rather than trying to learn protocols. One exception is the work of Kasai et al. [56], in which tabular Q-learning agents have to learn the content of a message to solve a predator-prey task with communication. Another example of open-ended communication learning in a multi-agent task is given in Giles and Jim [57]. Here evolutionary methods are used for learning the protocols which are evaluated on a similar predator-prey task. Their approach uses a fitness function that is carefully designed to accelerate learning. In general, heuristics and handcrafted rules have prevailed widely in this line of research. Moreover, typical tasks have been necessarily small so that global optimisation methods, such as evolutionary algorithms, can be applied. The use of deep representations and gradient-based optimisation as advocated in this work is an important departure, essential for scalability and further progress. A similar rationale is provided by Gregor et al. [58], another example of making an RL problem end-to-end differentiable.

Discrete communication

Unlike the recent work of Sukhbaatar et al. [59], we consider discrete communication channels and one of the key components of our methods is the signal binarisation during the decentralised execution. We believe that discreteness is a natural fit in studying the emergence of communication in different environments for the following reasons: First, in many environments the communication with other agents happens by acting in the environment. These actions can be discrete, while planning in these domains may require integrating over possible future actions. An example is the environment outlined in Section 2.6.1. Second, models with millions of continuous parameters can be difficult to interpret and understand. Discrete communication can help in the interpretability of the policies learnt, and can show traits like compositionality [60] similarly to language. Third, discreteness is an easy way to regulate the amount of information conveyed in a message, and also control the message robustness to noise as we show in our analysis in Section 2.6.3.

The idea of discrete communication is related to recent research on supervised learning and fitting neural networks in low-powered devices with memory and computational limitations using binary weights [e.g. 61], works on discovering binary codes for documents [62], and other discrete latent variable models. However, in a reinforcement learning setting the learning objective is different, and discritisation takes place in the encoding of messages exchanged between agents through a noisy channel. Finally, casting the problem as a multi-agent architecture instead of a single-agent problem, allows decentralised execution which is crucial in tackling a plethora of different settings (e.g. self-driving cars and computer network routing).

2.2.2 Background

Deep Q-Networks (DQN)

In a single-agent, fully-observable, RL setting [63], an agent observes the current state $s_t \in \mathcal{S}$ at each discrete time step t, chooses an action $u_t \in \mathcal{U}$ according to a

potentially stochastic policy π, observes a reward signal r_t, and transitions to a new state s_{t+1}. Its objective is to maximise an expectation over the discounted return,

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots , \tag{2.1}$$

where r_t is the reward received at time t and $\gamma \in [0,1]$ is a discount factor. The Q-function of a policy π is $Q^\pi(s,u) = \mathbb{E}\left[R_t | s_t = s, u_t = u\right]$. The optimal action-value function $Q^*(s,u) = \max_\pi Q^\pi(s,u)$ obeys the Bellman optimality equation

$$Q^*(s,u) = \mathbb{E}_{s'}\left[r + \gamma \max_{u'} Q^*(s',u') \mid s,u\right]. \tag{2.2}$$

Deep Q-learning [52] uses neural networks parameterised by θ to represent $Q(s,u;\theta)$. DQNs are optimised by minimising:

$$\mathcal{L}_i(\theta_i) = \mathbb{E}_{s,u,r,s'}[(y_i^{DQN} - Q(s,u;\theta_i))^2], \tag{2.3}$$

at each iteration i, with target

$$y_i^{DQN} = r + \gamma \max_{u'} Q(s',u';\theta_i^-). \tag{2.4}$$

Here, θ_i^- are the parameters of a target network that is frozen for a number of iterations while updating the online network $Q(s,u;\theta_i)$. The action u is chosen from $Q(s,u;\theta_i)$ by an *action selector*, which typically implements an ϵ-greedy policy that selects the action that maximises the Q-value with a probability of $1-\epsilon$ and chooses randomly with a probability of ϵ. DQN also uses *experience replay*: during learning, the agent builds a dataset of episodic experiences and is then trained by sampling mini-batches of experiences.

Independent DQN

DQN has been extended to cooperative multi-agent settings, in which each agent a observes the global s_t, selects an individual action u_t^a, and receives a team reward, r_t, shared among all agents. Tampuu et al. [64] address this setting with a framework that combines DQN with *independent Q-learning*, in which each agent a independently and simultaneously learns its own Q-function $Q^a(s,u^a;\theta_i^a)$. While

independent Q-learning can in principle lead to convergence problems (since one agent's learning makes the environment appear non-stationary to other agents), it has a strong empirical track record [65, 66], and was successfully applied to two-player pong.

Deep Recurrent Q-Networks

Both DQN and independent DQN assume full observability, i.e., the agent receives s_t as input. By contrast, in partially observable environments, s_t is hidden and the agent receives only an observation o_t that is correlated with s_t, but in general does not disambiguate it. Hausknecht and Stone [67] propose *deep recurrent Q-networks* (DRQN) to address single-agent, partially observable settings. Instead of approximating $Q(s,u)$ with a feed-forward network, they approximate $Q(o,u)$ with a recurrent neural network that can maintain an internal state and aggregate observations over time. This can be modelled by adding an extra input h_{t-1} that represents the hidden state of the network, yielding $Q(o_t,h_{t-1},u)$. For notational simplicity, we omit the dependence of Q on θ.

2.2.3 Setting

In this work, we consider RL problems with both multiple agents and partial observability. All the agents share the goal of maximising the same discounted sum of rewards R_t. While no agent can observe the underlying Markov state s_t, each agent a receives a private observation o_t^a correlated with s_t. In every time-step t, each agent selects an *environment action* $u_t^a \in U$ that affects the environment, and a *communication action* $m_t^a \in M$ that is observed by other agents but has no direct impact on the environment or reward. We are interested in such settings because it is only when multiple agents and partial observability coexist that agents have the incentive to communicate. As no communication protocol is given a priori, the agents must develop and agree upon such a protocol to solve the task.

Since protocols are mappings from action-observation histories to sequences of messages, the space of protocols is extremely high-dimensional. Automatically discovering effective protocols in this space remains an elusive challenge. In particular, the difficulty of exploring this space of protocols is exacerbated by the need for agents to coordinate the sending and interpreting of messages. For example, if one agent sends a useful message to another agent, it will only receive a positive reward if the receiving agent correctly interprets and acts upon that message. If it does not, the sender will be discouraged from sending that message again. Hence, positive rewards are sparse, arising only when sending and interpreting are properly coordinated, which is hard to discover via random exploration.

We focus on settings where communication between agents is not restricted during *centralised learning*, but during the *decentralised execution* of the learned policies, the agents can communicate only via a limited-bandwidth channel.

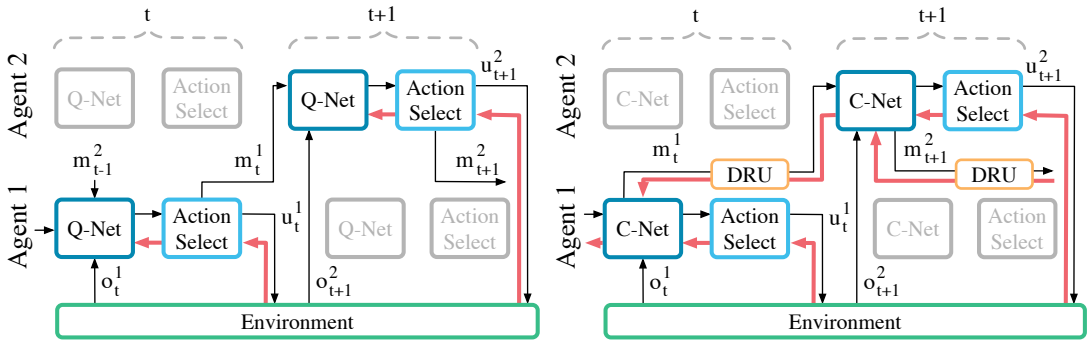
2.3 Reinforced Inter-Agent Learning (RIAL)

The most straightforward approach, which we call *reinforced inter-agent learning* (RIAL), is to combine DRQN with independent Q-learning for action and communication selection. Each agent's Q-network represents

$$Q^a(o_t^a, m_{t-1}^{a'}, h_{t-1}^a, u^a), \tag{2.5}$$

which conditions on that agent's individual hidden state h_{t-1}^a and observation o_t^a as well as messages from other agents $m_{t-1}^{a'}$.

To avoid needing a network with $|U||M|$ outputs, we split the network into Q_u^a and Q_m^a, the Q-values for the environment and communication actions, respectively. Similarly to [68], the action selector separately picks u_t^a and m_t^a from Q_u and Q_m, using an ϵ-greedy policy. Hence, the network requires only $|U| + |M|$ outputs and action selection requires maximising over U and then over M, but not maximising over $U \times M$.



(a) RIAL - RL based communication **(b)** DIAL - Differentiable communication

Figure 2.1: The bottom and top rows represent the communication flow for agent a_1 and agent a_2, respectively. In RIAL (a), all Q-values are fed to the action selector, which selects both environment and communication actions. Gradients, shown in red, are computed using DQN for the selected action and flow only through the Q-network of a single agent. In DIAL (b), the message m_t^a bypasses the action selector and instead is processed by the DRU (Section 5.2) and passed as a continuous value to the next C-network. Hence, gradients flow across agents, from the recipient to the sender. For simplicity, at each time step only one agent is highlighted, while the other agent is greyed out.

Both Q_u and Q_m are trained using DQN with the following two modifications, which were found to be essential for performance. First, we disable experience replay to account for the non-stationarity that occurs when multiple agents learn concurrently, as it can render experience obsolete and misleading. Second, to account for partial observability, we feed in the actions u and m taken by each agent as inputs on the next time-step. Figure 2.1(a) shows how information flows between agents and the environment, and how Q-values are processed by the action selector in order to produce the action, u_t^a, and message m_t^a. Since this approach treats agents as independent networks, the learning phase is not centralised, even though our problem setting allows it to be. Consequently, the agents are treated exactly the same way during decentralised execution as during learning.

Parameter Sharing

RIAL can be extended to take advantage of the opportunity for centralised learning by sharing parameters among the agents. This variation learns only one network, which is used by all agents. However, the agents can still behave differently

because they receive different observations and thus evolve different hidden states. In addition, each agent receives its own index a as input, allowing it to specialise. In our preliminary empirical evaluation we saw that excluding the agent's index resulted to a performance drop. The rich representations in deep Q-networks can facilitate the learning of a common policy while also allowing for specialisation. Parameter sharing also dramatically reduces the number of parameters that must be learned, thereby speeding learning. Under parameter sharing, the agents learn two Q-functions:

$$Q_u(o_t^a, m_{t-1}^{a'}, h_{t-1}^a, u_{t-1}^a, m_{t-1}^a, a, u_t^a), \tag{2.6}$$

$$Q_m(o_t^a, m_{t-1}^{a'}, h_{t-1}^a, u_{t-1}^a, m_{t-1}^a, a, u_t^a). \tag{2.7}$$

During decentralised execution, each agent uses its own copy of the learned network, evolving its own hidden state, selecting its own actions, and communicating with other agents only through the communication channel.

2.4 Differentiable Inter-Agent Learning (DIAL)

While RIAL can share parameters among agents, it still does not take full advantage of centralised learning. In particular, the agents do not give each other feedback about their communication actions. Contrast this with human communication, which is rich with tight feedback loops. For example, during face-to-face interaction, listeners send fast nonverbal queues to the speaker indicating the level of understanding and interest. RIAL lacks this feedback mechanism, which is intuitively important for learning communication protocols.

To address this limitation, we propose *differentiable inter-agent learning* (DIAL). The main insight behind DIAL is that the combination of centralised learning and Q-networks makes it possible, not only to share parameters but to push gradients from one agent to another through the communication channel. Thus, while RIAL is end-to-end trainable *within* each agent, DIAL is end-to-end trainable *across* agents. Letting gradients flow from one agent to another gives them richer

feedback, reducing the required amount of learning by trial and error, and easing the discovery of effective protocols.

DIAL works as follows: during centralised learning, communication actions are replaced with direct connections between the output of one agent's network and the input of another's. Thus, while the task restricts communication to discrete messages, during learning the agents are free to send real-valued messages to each other. Since these messages function as any other network activation, gradients can be passed back along the channel, allowing end-to-end backpropagation across agents.

In particular, the network, which we call a C-Net, outputs two distinct types of values, as shown in Figure 2.1(b), a) $Q(\cdot)$, the Q-values for the environment actions, which are fed to the action selector, and b) m_t^a, the real-valued vector message to other agents, which bypasses the action selector and is instead processed by the *discretise/regularise unit*, $\text{DRU}(m_t^a)$. The DRU regularises it during centralised learning, $\text{DRU}(m_t^a) = \text{Logistic}(\mathcal{N}(m_t^a, \sigma))$, where σ is the standard deviation of the noise added to the channel, and discretises it during decentralised execution, $\text{DRU}(m_t^a) = \mathbb{1}\{m_t^a > 0\}$. Figure 2.1 shows how gradients flow differently in RIAL and DIAL. The gradient chains for Q_u, in RIAL and Q, in DIAL, are based on the DQN loss. However, in DIAL the gradient term for m is the backpropagated error from the recipient of the message to the sender. Using this inter-agent gradient for training provides a richer training signal than the DQN loss for Q_m in RIAL. While the DQN error is nonzero only for the selected message, the incoming gradient is a $|m|$-dimensional vector that can contain more information. It also allows the network to directly adjust messages in order to minimise the downstream DQN loss, reducing the need for trial and error learning of good protocols.

While we limit our analysis to discrete messages, DIAL naturally handles continuous message spaces, as they are used anyway during centralised learning. At the same time, DIAL can also scale to large discrete message spaces, since it learns binary encodings instead of the one-hot encoding in RIAL, $|m| = O(\log(|M|)$.

2.5 Model architecture

RIAL and DIAL share the same neural network architecture. As illustrated in Figure 2.2, each agent is consists of a recurrent neural network (RNN) that maintains an internal state h, an input network for producing a task embedding z, and an output network for the Q-values and the messages m. The networks are unrolled for T time-steps.

The input for agent a is defined as a tuple of:

$$(o_t^a, m_{t-1}^{a'}, u_{t-1}^a, a). \tag{2.8}$$

The inputs a and u_{t-1}^a are passed through lookup tables, and $m_{t-1}^{a'}$ through a 1-layer MLP, both producing embeddings of size 128. o_t^a is processed through a task-specific network, TaskMLP, that produces an additional embedding of the same size. The

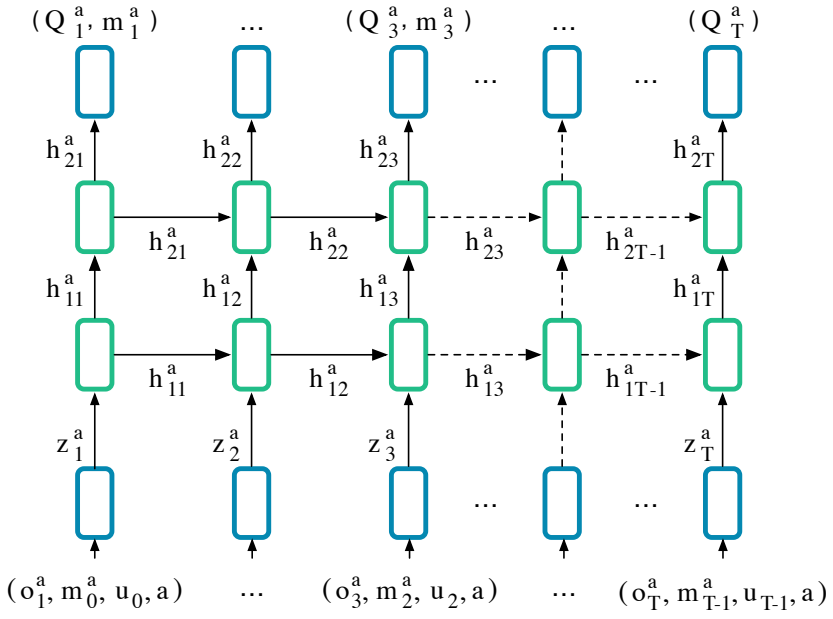


Figure 2.2: DIAL architecture. Agent a receives as input his id a, the task specific observation o_t^a, his last action u_{t-1}^a, and the communication channel m_{t-1}, and generates the state embedding $z_t^a = (\text{Lookup}(m) + \text{TaskMLP}(o_t^a) + \text{Lookup}(u_{t-1}^a) + \text{MLP}(m_{t-1}))$. Then, z_t^a is processed through a 2-layer RNN resulting $h_t^a = \text{GRU}[128, 128](z_t^a, h_{t-1}^a)$, which is used to approximate the agent's action-observation history. Finally, the output h_{2t}^a of the GRU is used at each step to compute $Q_t^a, m_t^a = \text{MLP}(h_{2t}^a)$.

state embedding is produced by element-wise summation of these embeddings,

$$z_t^a = \Big(\text{TaskMLP}(o_t^a) + \text{MLP}[|M|, 128](m_{t-1}) + \text{Lookup}(u_{t-1}^a) + \text{Lookup}(a)\Big). \tag{2.9}$$

We found that performance and stability improved when a batch normalisation layer [69] was used to preprocess m_{t-1}. z_t^a is processed through a 2-layer RNN with GRUs, $h_{1,t}^a = \text{GRU}[128, 128](z_t^a, h_{1,t-1}^a)$, which is used to approximate the agent's action-observation history. Finally, the output $h_{2,t}^a$ of the top GRU layer, is passed through a 2-layer MLP:

$$Q_t^a, m_t^a = \text{MLP}[128, 128, (|U| + |M|)](h_{2,t}^a). \tag{2.10}$$

Algorithm 1 formally describes DIAL. At each time-step, we pick an action for each agent ϵ-greedily with respect to the Q-function and assign an outgoing message

$$Q(\cdot), m_t^a = \text{C-Net}\left(o_t^a, \hat{m}_{t-1}^{a'}, h_{t-1}^a, u_{t-1}^a, a; \theta_i\right). \tag{2.11}$$

We feed in the previous action, u_{t-1}^a, the agent index, a, along with the observation o_t^a, the previous internal state, h_{t-1}^a and the incoming messages $\hat{m}_{t-1}^{a'}$ from other agents. After all agents have taken their actions, we query the environment for a state update and reward information.

When we reach the final time-step or a terminal state, we proceed to the backwards pass. Here, for each agent, a, and time-step, j, we calculate a target Q-value, y_j^a, using the observed reward, r_t, and the discounted target network. We then accumulate the gradients, $\nabla\theta$, by regressing the Q-value estimate

$$Q(o_t^a, \hat{m}_{t-1}^{a'}, h_{t-1}^a, u_{t-1}^a, a, u; \theta_i), \tag{2.12}$$

against the target Q-value, y_t^a, for the chosen action, u_t^a. We also update the message gradient chain μ_t^a, which contains the derivative of the downstream bootstrap error $\sum_{m,t'>t}\left(\Delta Q_{t+1}^{a'}\right)^2$ with respect to the outgoing message m_t^a.

To allow for efficient calculation, this sum can be broken into two parts. The first part, $\sum_{m'\neq m}\frac{\partial}{\partial \hat{m}_t^a}\left(\Delta Q_{t+1}^{a'}\right)^2$, captures the impact of the message on the total estimation error of the next step. The impact of the message m_t^a on all other

Algorithm 1 Differentiable Communication (DIAL)

Initialise θ_1 and θ_1^-
for each episode e **do**
  s_1 = initial state, $t = 0$, $h_0^a = \mathbf{0}$ for each agent a
  while $s_t \neq$ terminal **and** $t < T$ **do**
    $t = t + 1$
    for each agent a **do**
      Get messages $\hat{m}_{t-1}^{a'}$ of previous time-steps from
        agents m' and evaluate C-Net:
      $Q(\cdot), m_t^a = \text{C-Net}\left(o_t^a, \hat{m}_{t-1}^{a'}, h_{t-1}^a, u_{t-1}^a, a; \theta_i\right)$
      With probability ϵ pick random u_t^a, else
        $u_t^a = \max_a Q\left(o_t^a, \hat{m}_{t-1}^{a'}, h_{t-1}^a, u_{t-1}^a, a, u; \theta_i\right)$
      Set message $\hat{m}_t^a = \text{DRU}(m) = \begin{cases} \text{Logistic}(\mathcal{N}(m, \sigma)), \text{ if training} \\ \mathbb{1}\{m > 0\}, \text{ otherwise} \end{cases}$
    Get reward r_t and next state s_{t+1}
  Reset gradients $\nabla\theta = 0$
  for $t = T$ **to** 1, -1 **do**
    for each agent a **do**
      $y_t^a = \begin{cases} r_t, \text{ if } s_t \text{ terminal, else} \\ r_t + \gamma \max_u Q\left(o_{t+1}^a, \hat{m}_t^{a'}, h_t^a, u_t^a, a, u; \theta_i^-\right) \end{cases}$
      Accumulate gradients for action:
      $\Delta Q_t^a = y_t^a - Q\left(o_j^a, h_{t-1}^a, \hat{m}_{t-1}^{a'}, u_{t-1}^a, a, u_t^a; \theta_i\right)$
      $\nabla\theta = \nabla\theta + \frac{\partial}{\partial\theta}(\Delta Q_t^a)^2$
      Update gradient chain for differentiable communication:
      $\mu_j^a = \mathbb{1}\{t < T - 1\} \sum_{m' \neq m} \frac{\partial}{\partial \hat{m}_t^a}\left(\Delta Q_{t+1}^{a'}\right)^2 + \mu_{t+1}^{a'} \frac{\partial \hat{m}_{t+1}^{a'}}{\partial \hat{m}_t^a}$
      Accumulate gradients for differentiable communication:
      $\nabla\theta = \nabla\theta + \mu_t^a \frac{\partial}{\partial m_t^a} \text{DRU}(m_t^a) \frac{\partial m_t^a}{\partial \theta}$

  $\theta_{i+1} = \theta_i + \alpha\nabla\theta$
  Every C steps reset $\theta_i^- = \theta_i$

future rewards $t' > t + 1$ can be calculated using the partial derivative of the outgoing messages from the agents at time $t + 1$ with respect to the incoming message m_t^a, multiplied with their message gradients, $\mu_{t+1}^{a'}$. Using the message gradient, we can calculate the derivative with respect to the parameters, $\mu_t^a \frac{\partial \hat{m}_t^a}{\partial \theta}$.

Having accumulated all gradients, we conduct two parameter updates, first θ_i in the direction of the accumulated gradients, $\nabla\theta$, and then every C steps $\theta_i^- = \theta_i$. During decentralised execution, the outgoing activations in the channel are mapped into a binary vector, $\hat{m} = \mathbb{1}\{m_t^a > 0\}$. This ensures that discrete messages are exchanged, as required by the task.

In order to minimise the discretisation error when mapping from continuous values to discrete encodings, two measures are taken during centralised learning. First, Gaussian noise is added in order to limit the number of bits that can be encoded in a given range of m values. Second, the noisy message is passed through a logistic function to restrict the range available for encoding information. Together, these two measures regularise the information transmitted through the bottleneck. Furthermore, the noise also perturbs values in the middle of the range, due to the steeper slope, but leaves the tails of the distribution unchanged.

Formally, during centralised learning, m is mapped to $\hat{m} = \text{Logistic}\left(\mathcal{N}(m, \sigma)\right)$, where σ is chosen to be comparable to the width of the logistic function. In Algorithm 1, the mapping logic from m to $\hat{m}$ during training and execution is contained in the $\text{DRU}(m_t^a)$ function.

2.6 Experimental evaluation

In this section, we evaluate RIAL and DIAL with and without parameter sharing in two multi-agent problems and compare it with a no-communication shared-parameter baseline (NoComm). Results presented are the average performance across several runs, where those without parameter sharing (-NS), are represented by dashed lines. Across plots, rewards are normalised by the highest average reward achievable given access to the true state (Oracle). In our experiments, we use an ϵ-greedy policy with $\epsilon = 0.05$, the discount factor[1] is $\gamma = 1$, and the target network is

[1]In many domains there is no natural interpretation for the discount factor γ, the natural performance measure to optimise is the average reward received per time step [70, 71].

reset every 100 episodes. To stabilise learning, we execute parallel episodes in batches of 32. The parameters are optimised using RMSProp [72] with a learning rate of 5×10^{-4}. The architecture uses *rectified linear units* (ReLU), and *gated recurrent units* (GRU) [73], which have similar performance to *long short-term memory* [74] (LSTM) [75]. Unless stated otherwise, we set the standard deviation of noise added to the channel to $\sigma = 2$, which was found to be essential for good performance.

Generalisation

Although the hyper-parameters of our models were chosen carefully, they were selected to maximise the statistical performance of our models in the two environments of the experimental evaluation. In parallel, hyper-parameters such as episode length and the batch size were chosen to satisfy the computational hardware constraints. The immerse state-action space of these problems, e.g. the policy space of the switch riddle environment for 4 agents is 4^{354288} (see Section 2.6.1), makes overfitting extremely difficult, however, the generalisation of the chosen hyper-parameters over a large number of environments is an interesting challenge for future research.

Essential Tricks

We experimented using experience replay and found that using replay harms our performance because of the non-stationarity of the environment. We also find that adding noise to the channel is an essential feature in order to maintain performance after discretisation of the messages. Furthermore inputting the last action and using a batch normalisation layer to m help training.

2.6.1 Switch riddle

The first task is inspired by a well-known riddle described as follows: *"One hundred prisoners have been newly ushered into prison. The warden tells them that starting tomorrow, each of them will be placed in an isolated cell, unable*

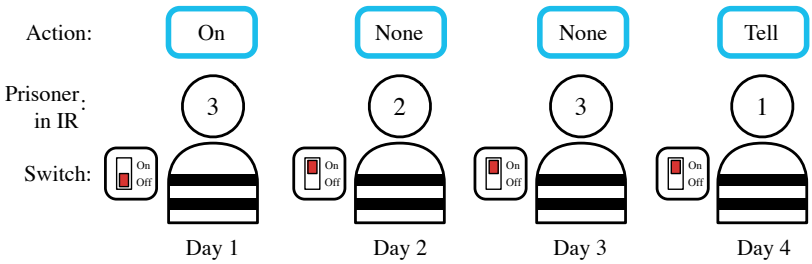


Figure 2.3: *Switch:* Every day one prisoner gets sent to the interrogation room where he sees the switch and chooses from "On", "Off", "Tell" and "None".

to communicate amongst each other. Each day, the warden will choose one of the prisoners uniformly at random with replacement, and place him in a central interrogation room containing only a light bulb with a toggle switch. The prisoner will be able to observe the current state of the light bulb. If he wishes, he can toggle the light bulb. He also has the option of announcing that he believes all prisoners have visited the interrogation room at some point in time. If this announcement is true, then all prisoners are set free, but if it is false, all prisoners are executed[...]" [76].

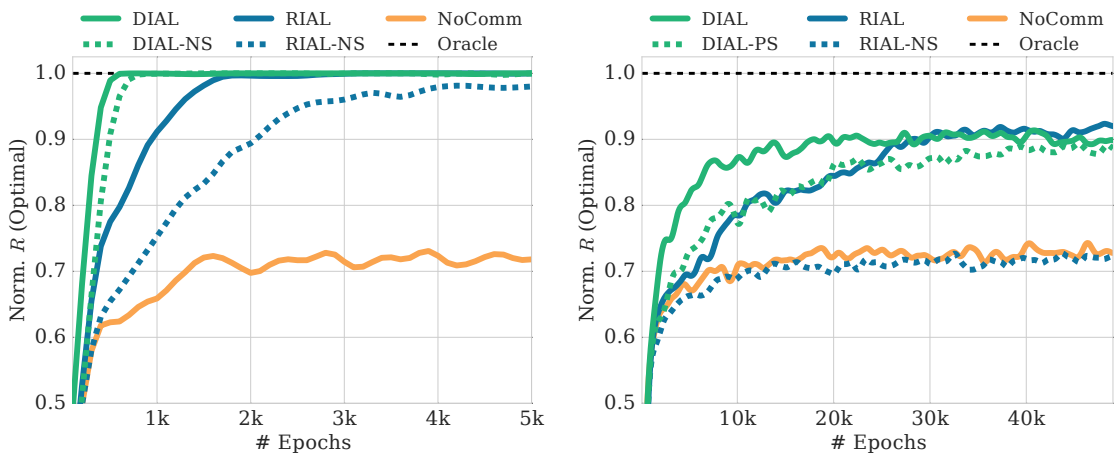
Architecture

In our formalisation, at time-step t, agent a observes $o_t^a \in \{0,1\}$, which indicates if the agent is in the interrogation room. Since the switch has two positions, it can be modelled as a 1-bit message, m_t^a. If agent a is in the interrogation room, then its actions are $u_t^a \in \{$"None","Tell"$\}$; otherwise the only action is "None". The episode ends when an agent chooses "Tell" or when the maximum time-step, T, is reached. The reward r_t is 0 unless an agent chooses "Tell", in which case it is 1 if all agents have been to the interrogation room and -1 otherwise. Following the riddle definition, in this experiment m_{t-1}^a is available only to the agent a in the interrogation room. Finally, we set the time horizon $T = 4n - 6$ in order to keep the experiments computationally tractable.

Complexity

The switch riddle poses significant protocol learning challenges. At any time-step t, there are $|o|^t$ possible observation histories for a given agent, with $|o| = 3$: the agent either is not in the interrogation room or receives one of two messages when it is. For each of these histories, an agent can chose between $4 = |U||M|$ different options, so at time-step t, the single-agent policy space is $(|U||M|)^{|o|^t} = 4^{3^t}$. The product of all policies for all time-steps defines the total policy space for an agent: $\prod 4^{3^t} = 4^{(3^{T+1}-3)/2}$, where T is the final time-step. The size of the multi-agent policy space grows exponentially in n, the number of agents: $4^{n(3^{T+1}-3)/2}$. We consider a setting where T is proportional to the number of agents, so the total policy space is $4^{n3^{O(n)}}$. For $n = 4$, the size is 4^{354288}. Our approach using DIAL is to model the switch as a continuous message, which is binarised during decentralised execution.

Experimental results



(a) Evaluation of $n = 3$ **(b)** Evaluation of $n = 4$

Figure 2.4: *Switch:* Performance of DIAL and RIAL, with and without (-NS) parameter sharing, and NoComm-baseline, for $n = 3$ and $n = 4$ agents.

Figure 2.4(a) shows our results for $n = 3$ agents. All four methods learn an optimal policy in 5k episodes, substantially outperforming the NoComm baseline. DIAL with parameter sharing reaches optimal performance substantially faster than RIAL. Furthermore, parameter sharing speeds both methods. Figure 2.4(b)

shows results for $n = 4$ agents. DIAL with parameter sharing again outperforms all other methods. In this setting, RIAL without parameter sharing was unable to beat the NoComm baseline. These results illustrate how difficult it is for agents to learn the same protocol independently. Hence, parameter sharing can be crucial for learning to communicate. DIAL-NS performs similarly to RIAL, indicating that the gradient provides a richer and more robust source of information.

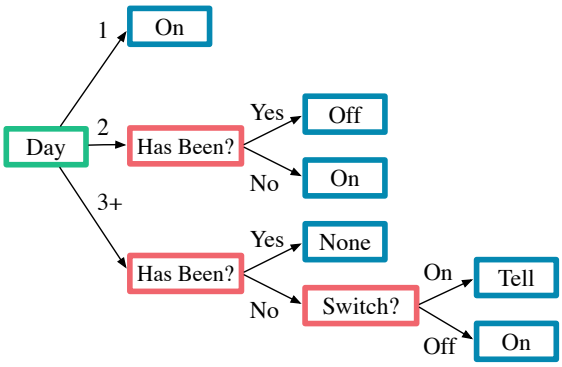


Figure 2.5: Protocol of $n = 3$

Figure 2.6: The decision tree extracted for $n = 3$ to interpret the communication protocol discovered by DIAL.

We also analysed the communication protocol discovered by DIAL for $n = 3$ by sampling 1K episodes, for which Figure 2.6 shows a decision tree corresponding to an optimal strategy. When a prisoner visits the interrogation room after day two, there are only two options: either one or two prisoners may have visited the room before. If three prisoners had been, the third prisoner would have finished the game. The other options can be encoded via the "On" and "Off" positions respectively.

2.6.2 MNIST games

In this section, we consider two tasks based on the well known MNIST digit classification dataset [77].

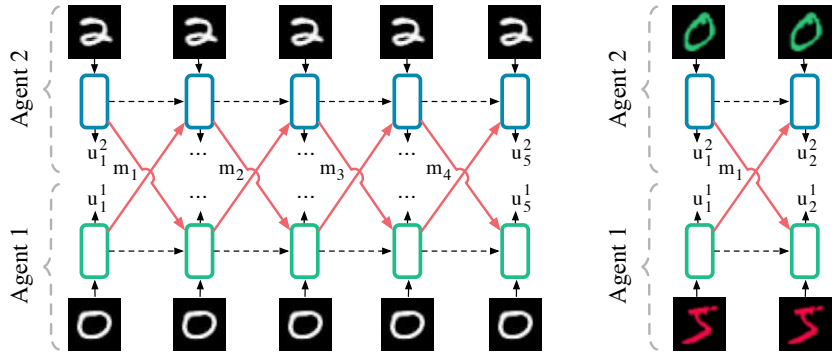


Figure 2.7: MNIST games architectures.

Colour-Digit MNIST

Colour-Digit MNIST is a two-player game in which each agent observes the pixel values of a random MNIST digit in red or green, while the colour label and digit value are hidden. The reward consists of two components that are antisymmetric in the action, colour, and parity of the digits. As only one bit of information can be sent, agents must agree to encode/decode either colour or parity, with parity yielding greater rewards. The game has two steps; in the first step, both agents send a 1-bit message, in the second step they select a binary action. The reward for each agent is $r(a) = 2(-1)^{a_2^a + c^a + d^{a'}} + (-1)^{a_2^a + d^a + c^{a'}}$ and the total cooperative reward is $r_2 = r(1) + r(2)$. In this task it is possible to change both the number of steps, t_f, of the game and the width of the communication channel, $|M|$. Setting the number of bits to lower than the minimum required in order to transmit the digit information forces the agents to communicate and integrate information over multiple time-steps. We present results for 5 time-steps with a 1-bit of information exchanged per step and for 2 time-steps with 4 bits exchanged per step.

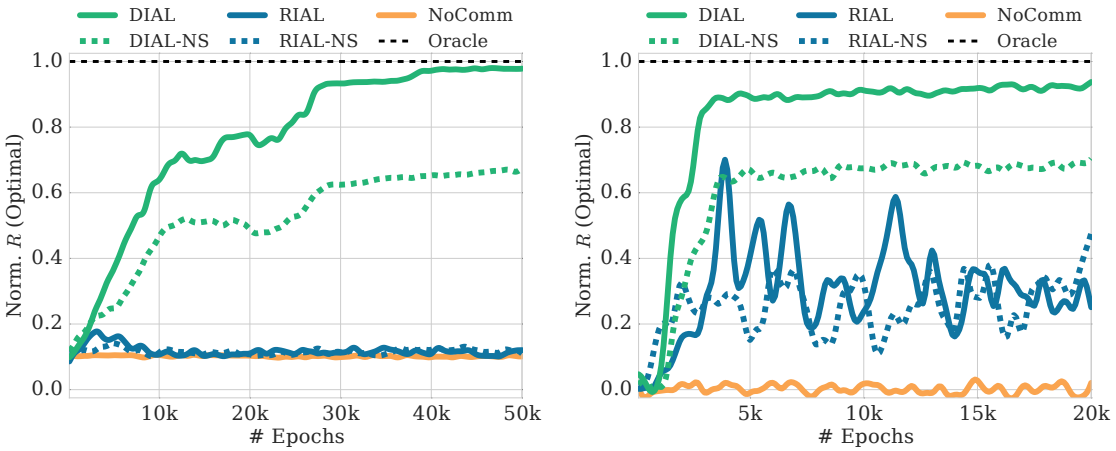
Multi-Step MNIST

Multi-step MNIST is a grayscale variant that requires agents to develop a communication protocol that integrates information across 5 time-steps in order to guess each others' digits. At each step, the agents exchange a 1-bit message and at the final step, $t = 5$, they are awarded $r = 0.5$ for each correctly guessed digit.

Architecture

The input processing network is a 2-layer MLP TaskMLP$[(|c|\times 28\times 28), 128, 128](o_t^a)$. Figure 2.7 depicts the generalised setting for both games. Our experimental evaluation showed improved training time using batch normalisation after the first layer.

Experimental results



(a) Evaluation of Multi-Step **(b)** Evaluation of Colour-Digit

Figure 2.8: *MNIST Games:* (a,b) Performance of DIAL and RIAL, with and without (-NS) parameter sharing, and NoComm, for both MNIST games.

Figures 2.8(a) and 2.8(b) show that DIAL substantially outperforms the other methods on both games. Furthermore, parameter sharing is crucial for reaching the optimal protocol. In multi-step MNIST, results were obtained with $\sigma = 0.5$. In this task, RIAL fails to learn, while in colour-digit MNIST it fluctuates around local minima in the protocol space; the NoComm baseline is stagnant at zero. DIAL's performance can be attributed to directly optimising the messages in order to reduce the global DQN error while RIAL must rely on trial and error. DIAL can also optimise the message content with respect to rewards taking place many time-steps later, due to the gradient passing between agents, leading to optimal performance in multi-step MNIST. To analyse the protocol that DIAL learned, we sampled 1K episodes. Figure 2.8(c) illustrates the communication bit sent at time-step t by agent 1, as a function of its input digit. Thus, each agent has learned a binary encoding

and decoding of the digits. These results illustrate that differentiable communication in DIAL is essential to fully exploiting the power of centralised learning and thus is an important tool for studying the learning of communication protocols.

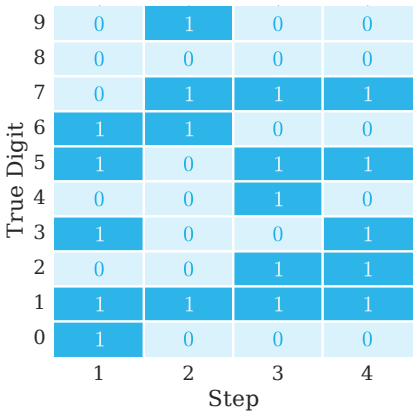


Figure 2.9: Extracted coding scheme for multi-step MNIST.

Our results show that DIAL deals more effectively with stochastic rewards in the *colour-digit MNIST* game than RIAL. To better understand why, consider a simpler two-agent problem with a structurally similar reward function $r = (-1)^{(s^1+s^2+a^2)}$, which is antisymmetric in the observations and action of the agents. Here random digits $s^1, s^2 \in 0, 1$ are input to agent 1 and agent 2 and $u^2 \in 1{,}2$ is a binary action. Agent 1 can send a single bit message, m^1. Until a protocol has been learned, the average reward for any action by agent 2 is 0, since averaged over s_1 the reward has an equal probability of being $+1$ or -1. Equally the TD error for agent 1, the sender, is zero for any message m:

$$\mathbb{E}\left[\Delta Q(s^1,m^1)\right] = Q(s^1,m^1) - \mathbb{E}\left[r(s^2,a^2,s^1)\right]_{s^2,a^2} = 0 - 0, \tag{2.13}$$

By contrast, DIAL allows for learning. Unlike the TD error, the gradient is a function of the action and the observation of the receiving agent, so summed across different $+1/-1$ outcomes the gradient updates for the message m no longer cancel:

$$\mathbb{E}\left[\nabla\theta\right] = \mathbb{E}\left[\left(Q(s^2,m^1,a^2) - r(s^2,a^2,s^1)\right)\frac{\partial}{\partial m}Q(s^2,m^1,a^2)\frac{\partial}{\partial\theta}m^1(s^1)\right]_{<s^2,a^2>}. \tag{2.14}$$

2.6.3 Effect of channel noise

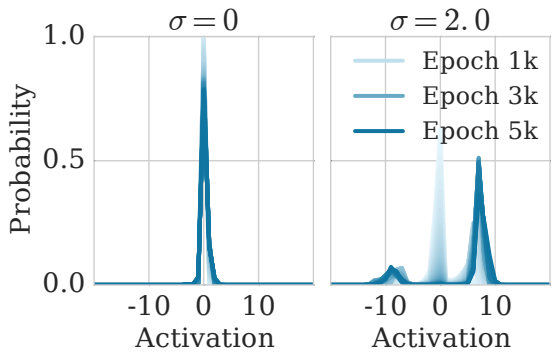


Figure 2.10: DIAL's learned activations with and without noise in DRU.

The question of why language evolved to be discrete has been studied for centuries, see e.g., the overview in [78]. Since DIAL learns to communicate in a continuous channel, our results offer an illuminating perspective on this topic. In particular, Figure 2.10 shows that, in the switch riddle, DIAL without noise in the communication channel learns centred activations. By contrast, the presence of noise forces messages into two different modes during learning. Similar observations have been made in relation to adding noise when training document models [62] and performing classification [61]. In our work, we found that adding noise was essential for successful training.

Given that the amount of noise, σ, is a hyperparameter that needs to be set, it is useful to understand how it impacts the amount of information that can pass through the channel. A first intuition can be gained by looking at the width of the sigmoid: Taking the decodable range of the logistic function to be x values corresponding to y values between 0.01 and 0.99, an initial estimate for the range is ≈ 10. Thus, requiring distinct x values to be at least six standard deviations apart, with $\sigma = 2$, only two bits can be encoded reliably in this range. To get a better understanding of the required σ we can visualise the capacity of the channel including the logistic function and the Gaussian noise. To do so, we must

first derive an expression for the probability distribution of outgoing messages, $\hat{m}$, given incoming activations, m, $P(\hat{m}|m)$:

$$P(\hat{m}|m) = \frac{1}{\sqrt{2\pi}\sigma\hat{m}(1-\hat{m})} \exp\left(-\frac{\left(m - \log(\frac{1}{\hat{m}} - 1)\right)^2}{\sigma^2}\right). \tag{2.15}$$

For any m, this captures the distribution of messages leaving the channel. Two m values m_1 and m_2 can be distinguished when the outgoing messages have a small probability of overlapping. Given a value m_1 we can thus pick a next value m_2 to be distinguishable when the highest value $\hat{m}_1$ that m_1 is likely to produce is less than the lowest value $\hat{m}_2$ that m_2 is likely to produce. An approximation for when this happens is when $(\max_{\hat{m}} s.t. P(\hat{m}|m_1) > \epsilon) = (\min_{\hat{m}} s.t. P(\hat{m}|m_2) > \epsilon)$. Figure 2.11 illustrates this for three different values of σ. For $\sigma > 2$, only two options can be reliably encoded using $\epsilon = 0.1$, resulting in a channel that effectively transmits only one bit of information.

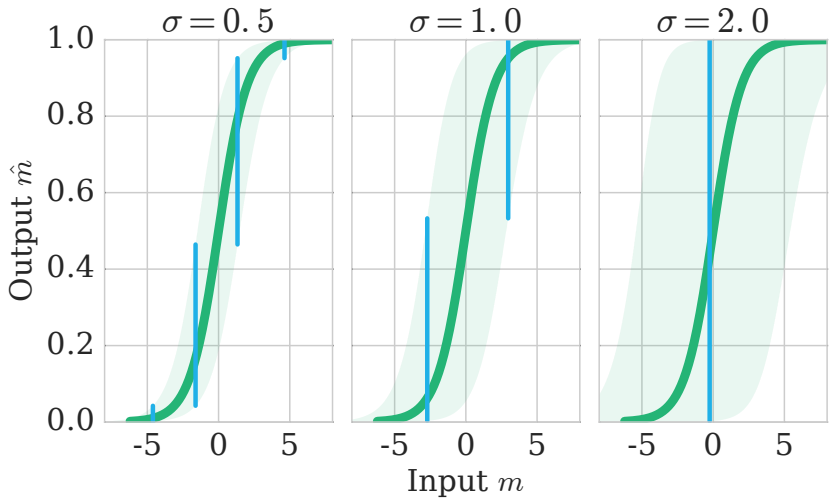


Figure 2.11: Distribution of regularised messages, $P(\hat{m}|m)$ for different noise levels. Shading indicates $P(\hat{m}|m) > 0.1$. Blue bars show a division of the x-range into intervals s.t. the resulting y-values have a small probability of overlap, leading to decodable values.

Interestingly, the amount of noise required to regularise the channel depends greatly on the benefits of over-encoding information. More specifically, as illustrated in Figure 2.12, in tasks where sending more bits does not lead to higher rewards,

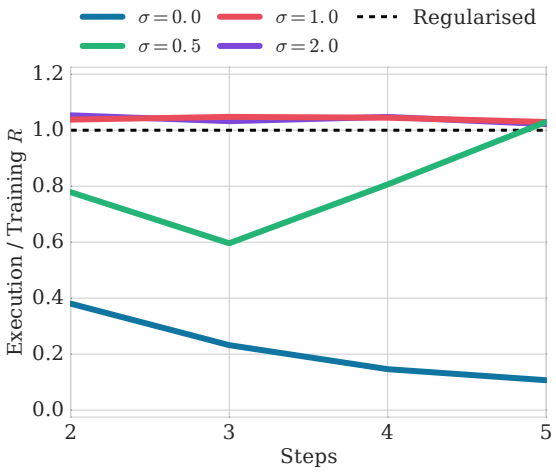


Figure 2.12: Final evaluation performance on multi-step MNIST of DIAL normalised by training performance after 50K epochs, under different noise regularisation levels $\sigma \in \{0, 0.5, 1, 1.5, 2\}$, and different numbers of steps $step \in [2, \ldots, 5]$.

small amounts of noise are sufficient to encourage discretisation, as the network can maximise reward by pushing activations to the tails of the sigmoid, where the noise is minimised. The figure illustrates the final average evaluation performance normalised by the training performance of three runs after 50K of the multi-step MNIST game, under different noise regularisation levels $\sigma \in \{0, 0.5, 1, 1.5, 2\}$, and different numbers of steps $step \in [2, \ldots, 5]$. When the lines exceed "Regularised", the test reward, after discretisation, is higher than the training reward, i.e., the channel is properly regularised and getting used as a single bit at the end of learning. Given that there are 10 digits to encode, four bits are required to get full reward. Reducing the number of steps directly reduces the number of bits that can be communicated, $\#bits = steps - 1$, and thus creates an incentive for the network to over-encode information in the channel, which leads to greater discretisation error. This is confirmed by the normalised performance for $\sigma = 0.5$, which is around 0.7 for 2 steps (1 bit) and then goes up to > 1 for 5 steps (4 bits). Note also that, without noise, regularisation is not possible and that with enough noise the channel is always regularised, even if over-encoding information would yield higher training rewards.

2.7 Conclusion

This work advanced novel environments and successful techniques for learning communication protocols and studying the emergence of communication. It presented a detailed comparative analysis covering important factors involved in the learning of communication protocols with deep networks, including differentiable communication, neural network architecture design, channel noise, tied parameters, and other methodological aspects.

It should be seen as a first attempt at learning communication and language with deep learning approaches. The gargantuan task of understanding communication and language in their full splendour, covering compositionality, concept lifting, conversational agents, and many other important problems still lies ahead. We are however optimistic that the approaches proposed in this work can help tackle these challenges. Finally, to facilitate further research in the field the source code of our setup is available at: `https://github.com/iassael/learning-to-communicate`.

Finally, in this chapter we have shown that such protocols can be learnt using reinforcement learning, and how the learning process can be improved by introducing differentiable communication and end-to-end training. The next chapter focuses on the subprocess recognising messages, and more specifically in the case of human verbal communication.

3 Recognition

Having studied the emergence of communication, we focus on the recognition of verbal communication, and more specifically speech recognition for visual verbal communication, also know as lipreading. Lipreading is the task of decoding text from the movement of a speaker's mouth. This chapter studies whether by using deep learning sentence-level lipreading can be tackled more effectively, and how could it positively impact the experience of patients with speech impairments. We present two state-of-the-art methods LipNet, and Vision-to-Phoneme (V2P). LipNet is a model that maps videos of lips to sequences of character distributions, making use of spatiotemporal convolutions, a recurrent network, and the connectionist temporal classification loss, trained entirely end-to-end. To our knowledge, LipNet is the first end-to-end sentence-level lipreading model. V2P is our follow-up work presenting an improved large-scale system, consisting of a video processing pipeline that maps raw video to stable videos of lips, a scalable deep neural network that maps the lip videos to sequences of phoneme distributions, and a production-level speech decoder that outputs sequences of words. To train V2P we construct the largest existing real-world lipreading dataset, consisting of pairs of transcriptions and video clips of faces speaking. V2P significantly improves on previous lipreading approaches, including variants of LipNet and of Watch, Attend, and Spell (WAS), and thus this chapter will focus on V2P.

3.1 Introduction

Deep learning techniques have allowed for significant advances in recognition of verbal communication such as lipreading over the last few years [3, 79–83]. However, these approaches have often been limited to narrow vocabularies, and relatively small datasets [3, 80, 83]. Often the approaches focus on single-word classification [28, 84–96] and do not attack the open-vocabulary continuous recognition setting. In this work, we contribute a novel method for large-vocabulary continuous visual speech recognition. In contrast to the largest previously reported lipreading vocabulary, 17,428 terms, we report substantial reductions in word error rate (WER) over the state-of-the-art approaches to lipreading even with a larger vocabulary of 127,055 terms.

Assisting people with speech impairments is the motivating factor behind this work. Visual speech recognition could positively impact the lives of hundreds of thousands of patients with speech impairments worldwide. For example, in the U.S. alone 103,925 tracheostomies were performed in 2014 [97], a procedure that can result in a difficulty to speak (dysphonia) or an inability to produce voiced sound (aphonia). While this work focuses on developing a scalable solution to lipreading using a vast diverse dataset, we also expand on this important medical application in Section 3.2. The discussion there has been provided by medical experts and is aimed at medical practitioners.

We propose a novel lipreading system, illustrated in Figure 3.1, which transforms raw video into a word sequence. The first component of this system is a data processing pipeline used to create the largest existing visual speech recognition dataset, distilled from YouTube videos, consisting of phoneme sequences paired with video clips of faces speaking (3,886 hours of video). The creation of the dataset alone required a non-trivial combination of computer vision and machine learning techniques. At a high-level this process takes as input raw video and annotated audio segments, filters and preprocesses them, and produces a collection

of aligned phoneme and lip frame sequences. The details of this process are described in Section 3.4.

Next, this work introduces a new neural network architecture for lipreading, which we call *Vision to Phoneme* (V2P), trained to produce a sequence of phoneme distributions given a sequence of video frames. In light of the large scale of our dataset, the network design has been highly tuned to maximise predictive performance subject to the strong computational and memory limits of modern GPUs. Our approach is the first to combine a deep learning-based phoneme recognition model with production-grade word-level decoding techniques. By decoupling phoneme prediction and word decoding as is often done in speech recognition, we are able to arbitrarily extend the vocabulary without retraining the neural network. Details of our model and this decoding process are given in Section 3.5. By design, the trained model only performs well when videos are shot at specific angles when a subject is facing the camera, within a certain distance from a subject, and at high quality. It does not perform well in other contexts.

Finally, this entire lipreading system results in an unprecedented WER of 40.9% as measured on a held-out set from our dataset. In comparison, professional lipreaders achieve either 86.4% or 92.9% WER on the same dataset, depending on the amount of context given. Similarly, previous approaches such as variants of LipNet [3] and of *Watch, Attend, and Spell* (WAS) [79] demonstrated WERs of only 89.8% and 76.8% respectively.

3.2 Medical motivation

As a consequence of injury or disease and its associated treatment, millions of people worldwide have communication problems preventing them from generating sound. As hearing aids and cochlear transplants have transformed the lives of people with hearing loss, there is potential for lip reading technology to provide alternative communication strategies for people who have lost their voice.

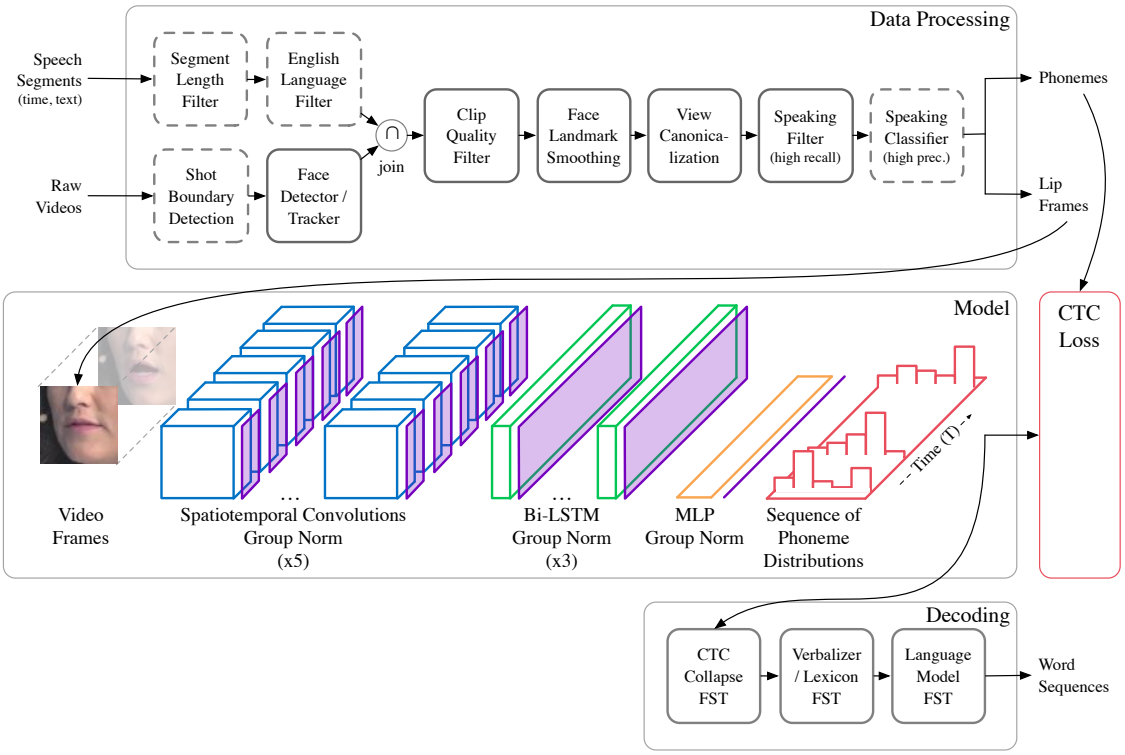


Figure 3.1: The full visual speech recognition system introduced by this work consists of a data processing pipeline that generates lip and phoneme clips from YouTube videos (see Section 3.4), and a scalable deep neural network for phoneme recognition combined with a production-grade word-level decoding module used for inference (see Section 3.5).

3.2.1 Aphonia

Aphonia is the inability to produce voiced sound. It may result from injury, paralysis, removal or other disorders of the larynx. Common examples of primary aphonia include bilateral recurrent laryngeal nerve damage as a result of thyroidectomy *(removal of the thyroid gland and any tumour)* for thyroid cancer, laryngectomy *(surgical removal of the voice box)* for laryngeal cancers, or tracheostomy *(the creation of an alternate airway in the neck bypassing the voicebox).*

3.2.2 Dysphonia

Dysphonia is difficulty in speaking due to a physical disorder of the mouth, tongue, throat, or vocal cords. Unlike aphonia, patients retain some ability to speak.

For example, in Spasmodic dysphonia, a disorder in which the laryngeal muscles go into periods of spasm, patients experience breaks or interruptions in the voice, often every few sentences, which can make a person difficult to understand.

We see this work having potential medical applications for patients with aphonia or dysphonia in at least two distinct settings. Firstly, an acute care setting (i.e. a hospital with an emergency room and an intensive care unit), patients frequently undergo elective (planned) or emergency (unplanned) procedures (e.g. Tracheostomy) which may result in aphonia or dysphonia. In the U.S. 103,925 tracheostomies were performed in 2014, resulting in an average hospital stay of 29 days [97]. Similarly, in England and Wales 15,000 tracheostomies are performed each year [98].

Where these procedures are unplanned, there is often no time or opportunity to psychologically prepare the patient for their loss of voice, or to teach the patient alternative communication strategies. Some conditions that necessitate tracheotomy, such as high spinal cord injuries, also affect limb function, further hampering alternative communication methods such as writing.

Even where procedures are planned, such as for head and neck cancers, despite preparation of the patient through consultation with a speech and language therapist, many patients find their loss of voice highly frustrating especially in the immediate post-operative period.

Secondly, where surgery has left these patients cancer-free, they may live for many years, even decades without the ability to speak effectively, in these patients we can envisage that they may use this technology in the community, after discharge from hospital. While some patients may either have tracheotomy reversed, or adapt to speaking via a voice prosthesis, electro-larynx or esophageal speech, many patients do not achieve functional spoken communication. Even in those who achieve good face-to-face spoken communication, few laryngectomy patients can communicate effectively on the telephone, and face the frequent frustration of being hung-up on by call centres and others who do not know them.

3.2.3 Acute care applications

It is widely acknowledged that patients with communication disabilities, including speech impairment or aphonia can pose significant challenges in the clinical environment, especially in acute care settings, leading to potentially poorer quality of care [99]. While some patients will be aware prior to surgery that they may wake up unable to speak, for many patients in the acute setting (e.g. Cervical Spinal Cord Injury, sudden airway obstruction) who wake up following an unplanned tracheotomy, their sudden inability to communicate can be phenomenally distressing.

3.2.4 Community applications

Patients who are discharged from hospital without the ability to speak, or with poor speech quality, face a multitude of challenges in day-to-day life which limits their independence, social functioning and ability to seek employment.

We hypothesise that the application of technology capable of lip-reading individuals with the ability to move their facial muscles, but without the ability to speak audibly could significantly improve quality of life for these patients. Where the application of this technology improves the person's ability to communicate over the telephone, it would enhance not only their social interactions, but also their ability to work effectively in jobs that require speaking over the phone.

Finally, in patients who are neither able to speak, nor to move their arms, this technology could represent a step-change in terms of the speed at which they can communicate, as compared to eye-tracking or facial muscle based approaches in use today.

3.2.5 Potential impact

The aforementioned applications could be of high-impact for the large body of the population with speech impairments. Only in the U.S. 103,925 tracheostomies were performed in 2014 [97], a procedure that can result in dysphonia or aphonia.

After consulting experts and visiting acute care units in the United Kingdom, we saw that the majority of those patients communicate with the doctors using either positive or negative signals by head or body movements, or by employing professional lipreader services. Unarguably, physical binary coding of answers is a very inefficient way of communication, and the patient's replies are bound to the questions being asked. On the other hand, professional lipreaders are quite expensive and thus such services if used are employed only for a few hours. An interesting observation was that people or family members of a patient may sometimes perform comparably good to professional services showing the importance of context.

All aforementioned communication solutions are inefficient and suboptimal. At the same time, it will be very hard for visual speech recognition models to even approach the error rates of audio speech recognition because of the nature of the task. Although roughly speaking error rates below 10% make speech recognition effective, we believe that any improvement in the current state would be of great benefit. Even improving the accuracy by limiting the predictions to a very small vocabulary of important words would allow still more options compared to binary coding and at the same time it would be more accessible than a professional lipreader. Thus, by introducing models such as LipNet and V2P that tackle the task of visual speech recognition, and by improving on their error rates, we could hopefully help the communication and the experience of these patient groups.

3.3 Automated visual speech recognition

3.3.1 Background

While there is a large body of literature on automated lipreading, much of the early work focused on single-word classification and relied on substantial prior knowledge [100–107]. For example, Goldschen et al. [108] predicted continuous sequences of tri-visemes using a traditional HMM model with visual features extracted from a codebook of clustered mouth region images. The predicted

visemes were used to distinguish sentences from a set of 150 possible sentences. Furthermore, Potamianos et al. [109] predict words and sequences digits using HMMs, Potamianos and Graf [110] introduce multi-stream HMMs, and Potamianos et al. [111] improve the performance by using visual features in addition to the lip contours. Later, Chu and Huang [100] used coupled HMMs to jointly model audio and visual streams to predict sequences of digits. Neti et al. [112] used HMMs for sentence-level speech recognition in noisy environments, using the IBM ViaVoice dataset, by fusing handcrafted visual and audio features. More recent attempts using traditional speech, vision and machine learning pipelines include the works of Gergen et al. [113], Paleček [114], Hassanat [115] and Bear and Harvey [116]. For further details, we refer the reader to the survey material of Potamianos et al. [117] and Zhou et al. [118].

However, as noted by Zhou et al. [118] and Assael* et al. [3], until recently generalisation across speakers and extraction of motion features have been considered open problems. Advances in deep learning have made it possible to overcome these limitations, but most works still focus on single-word classification, either by learning visual-only representations [28, 84–86, 96], multimodal audio-visual representations [87–91], or combining deep networks with traditional speech techniques (e.g. HMMs and GMM-HMMs) [92–95].

3.3.2 Sentence-level lipreading using deep learning

LipNet [3] was the first end-to-end model to tackle sentence-level lipreading by predicting character sequences. The model combined spatiotemporal convolutions with gated recurrent units (GRUs) and was trained using the CTC loss function. Figure 3.2 illustrates the LipNet architecture, which starts with 3×(spatiotemporal convolutions, channel-wise dropout, spatial max-pooling). Subsequently, the features extracted are followed by two Bi-GRUs. The Bi-GRUs are crucial for efficient further aggregation of the STCNN output. Finally, a linear transformation is applied at each time-step, followed by a softmax over the vocabulary augmented with the

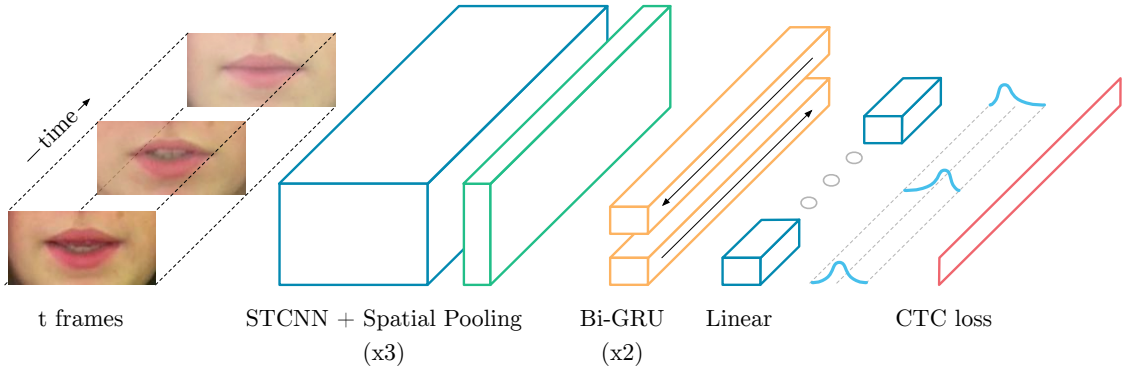


Figure 3.2: LipNet architecture. A sequence of T frames is used as input, and is processed by 3 layers of STCNN, each followed by a spatial max-pooling layer. The features extracted are processed by 2 Bi-GRUs; each time-step of the GRU output is processed by a linear layer and a softmax. This end-to-end model is trained with CTC.

CTC blank, and then the CTC loss. All layers use rectified linear unit (ReLU) activation functions. More details including hyperparameters can be found in Table 3.4. LipNet was evaluated on the GRID corpus [119], a limited grammar and vocabulary dataset consisting of 28 hours of 5-word sentences, where it achieved 4.8% and 11.4% WER in overlapping and unseen speaker evaluations respectively. By comparison, the performance of competent human lipreaders on GRID was 47.7%. LipNet is the closest model to our neural network.

Several similar architectures were subsequently introduced in the works of Thanda and Venkatesan [80] who study audio-visual feature fusion, Koumparoulis et al. [81] who work on a small subset of 18 phonemes and 11 words to predict digit sequences, and Xu et al. [83] who presented a model cascading CTC with attention. Chung et al. [79] were the first to use sequence-to-sequence models with attention to tackle audio-visual speech recognition with a real-world dataset. The model "Watch, Listen, Attend and Spell" (WLAS), consists of a visual (WAS) and an audio (LAS) module. To evaluate WLAS, the authors created LRS, the largest dataset at that point with approximately 246 hours of clips from BBC news broadcasts, and introduced an efficient video processing pipeline to generate the dataset. The authors reported 50.2% WER, with the performance of professional lipreaders being 87.6% WER. Chung and Zisserman [82] extended the work to

multi-view sentence-level lipreading, achieving 62.8% WER for profile views and 56.4% WER for frontal views. Both Chung et al. [79] and Chung and Zisserman [82] pre-learn features with the audio-video synchronization classifier of Chung and Zisserman [120], and fix these features in order to compensate for the large memory requirements of their attention networks. Other related advances include works using vision for silent speech reconstruction [121–124] and for separating an audio signal into its individual speech sources [125, 126].

In contrast to LipNet Assael* et al. [3] our model, V2P, uses a network to predict a sequence of phoneme distributions which are then fed into a decoder to produce a sequence of words. This flexible design enables us to easily accommodate very large vocabularies, and in fact we can extend the size of the vocabulary without having to retrain the deep network. Unlike previous work, V2P is memory and computationally efficient without requiring pre-trained features [79, 82].

3.4 Building a data pipeline for large-scale visual speech recognition

In this section we discuss the data processing pipeline, again illustrated in Figure 3.1, used to create the *Large-Scale Visual Speech Recognition* (LSVSR) dataset used in this work. The result of this pipeline is a significantly larger and more diverse dataset than in all previous efforts. While the first large-vocabulary lipreading dataset was IBM ViaVoice [112], more recent work has resulted in the much larger LRS and MV-LRS[1] datasets [79, 82], both generated from BBC news broadcasts. However, LSVSR is an order of magnitude greater than any previous dataset with 3,886 hours of audio-video-text pairs. In addition, the content is much more varied (i.e. not news-specific), resulting in a 7.3× larger vocabulary of

[1]MV-LRS is the only publicly available large-vocabulary dataset, however it is limited to academic usage.

127,055 words. Table 3.1 shows a comparison of sentence-level (word sequence) visual speech recognition datasets.

Dataset	Utter.	Hours	Vocab
GRID [119]	33,000	28	51
IBM ViaVoice [112]	17,111	35	10,400
MV-LRS [82]	74,564	∼155	14,960
LRS [79]	118,116	∼246	17,428
LSVSR (Ours)	**2,934,899**	**3,886**	**127,055**

Table 3.1: A comparison of sentence-level (word sequence) visual speech recognition datasets.

For further comparison with LRS, Figure 3.3 shows the frequency of words in the LSVSR dataset in decreasing order of occurrence; approximately 350K words occur at least 3 times. This histogram was used to select a vocabulary of 127,055 words as it captures most of the mass. As it can be seen from the figure, thresholding at a vocabulary of 17,428 words, as LRS, the largest existing previous dataset, excludes a large portion of high probability words.

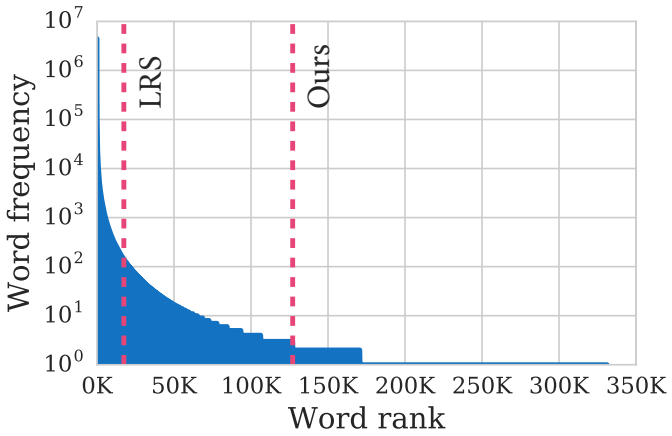


Figure 3.3: Frequency of words in the LSVSR dataset in decreasing order of occurrence; approximately 350K words occur at least 3 times. We used this histogram to select a vocabulary of 127,055 words as it captures most of the mass. Note that thresholding at a vocabulary of 17,428 words, the largest existing previous dataset, excludes a large portion of high probability words.

Our pipeline makes heavy use of large-scale parallel processing and is imple-

mented as a number of independent modules and filters on top of FlumeJava [127]. This pipeline takes as input raw video and speech segments and outputs paired sequences of phonemes and lip frames which can be used to train a phoneme model described in Section 3.5. By eliminating the components marked by dashes in Figure 3.1, i.e. those components whose primary use are in producing paired training data, this same pipeline can be used in combination with a trained model to predict word sequences from raw videos.

Our dataset is extracted from public YouTube videos. This is a common strategy for building datasets in ASR and speech enhancement [128–131, 125]. In our case, we use the work of Liao et al. [128] to extract audio clips paired with transcripts, yielding 140,000 hours of audio segments. Our processing pipeline is built on top of that, beginning with using the audio segments to fetch corresponding video segments.

3.4.1 Length filter, language filter

The duration of each segment extracted from YouTube is limited to between 1 and 12 seconds, and the transcripts are filtered through a language classifier [132] to remove non-English utterances. For evaluation, we further remove the utterances containing fewer than 6 words. Finally, the aligned phoneme sequences are obtained using a standard forced alignment approach with a lexicon with multiple pronunciations [128]. The phonetic alphabet is a reduced version of X-SAMPA [133] with 40 phonemes plus silence.

3.4.2 Raw videos, shot boundary detection, face detection

First, constant spatial padding in each video segment is eliminated. Then a standard, thresholding colour histogram classifier [134] identifies segments containing shot boundaries and removes them. Finally, FaceNet [135] detects and tracks faces in every remaining segment.

3.4.3 Clip quality filter

Here, speech segments are joined with the set of tracked faces identified in their corresponding videos. We then filter based on the quality of the video: we remove blurry clips, clips with faces with an eye-to-eye width of less than 80 pixels (i.e. low resolution videos and clips where the face occupies little of the frame), and frame rates lower than 23fps. We allow a range of frame rates as input as varying frame rates has a similar effect as peoples' different speaking paces; however, frame rates above 30fps are downsampled.

3.4.4 Face landmark smoothing

The segments are then processed by a face landmark tracker and the resulting landmark positions are smoothed using a temporal Gaussian kernel. Empirically, our preliminary studies showed smoothing was crucial for achieving optimal performance. Next, following previous literature [79], we keep segments where the face yaw and pitch remain within $-30°$ and $30°$. Models trained outside this range perform worse [82].

3.4.5 View canonicalisation

We obtain canonical faces using a reference canonical face model and by applying an affine transformation on the landmarks. Then, we use a thumbnail extractor which is configured to crop the area around the lips of the canonical face.

3.4.6 Speaking filter

Using the extracted and smoothed landmarks, minor lip movements and non-speaking faces are discarded using a threshold filter. This process involves computing the mouth openness in all frames, normalising by the size of the face bounding box, and then thresholding on the standard deviation of the normalised openness.

This classifier has very low computational cost, but a high recall, e.g. voice-overs are not handled.

3.4.7 Speaking classifier

As a final step, we build V2P-Sync, a neural network architecture to verify the audio and video channel alignment inspired by the work of Chung and Zisserman [120] and Torfi et al. [136]. V2P-Sync uses longer time segments as inputs and spatiotemporal convolutions as compared to the spatial-only convolutions of Chung and Zisserman, and landmark smoothing and view canonicalisation as compared to Torfi et al.. These characteristics facilitate the extraction of temporal features which is key to our task. Our model, V2P-Sync, takes as input a pair of a log mel-spectrogram and 9 grayscale video frames and produces an embedding for each using two separate neural network architectures. If the Euclidean distance of the audio and video embeddings is less than a given threshold then the pair is classified as synchronized. The architecture is trained using a contrastive loss similar to Chung and Zisserman. Since there is no labeled data for training, the initial unfiltered pairs are used as positive samples with negative samples generated by randomly shifting the video of an unfiltered pair. After convergence, the dataset is filtered using the trained model, which is then fine-tuned on the resulting subset of the initial dataset. The final model is used to filter the dataset a second time, achieving an accuracy of 81.2%. This accuracy is improved as our audio-video pairs are processed by sliding V2P-Sync on 100 equally spaced segments and their scores are averaged. The V2P-Sync networks in Tables 3.2 and 3.3 are optimised using a batch size of 128, batch normalisation, and Adam [137] with a learning rate of 10^{-4} and default hyperparameters: first and second momentum coefficients 0.9 and 0.999 respectively, and $\epsilon = 10^{-8}$ for numerical stability.

Finally, by combining all of these components we obtain a dataset consisting of paired video and phoneme sequences, where video sequences are represented as identically-sized frames (here, 128×128) stacked in the time-dimension. Our

pipeline processed clips pre-selected from YouTube using the work of Liao et al. [128], but only about 2% of clips satisfied the filtering criteria (face detection, frame rate, resolution, blur, face rotation) and had lip movements matched with text as determined by our speaking classifier.

Table 3.2: V2P-Sync video embedding neural network architecture.

Layer	Filter size	Stride	Output channels	Input
conv1	$3 \times 3 \times 3$	$1 \times 2 \times 2$	16	$9 \times 128 \times 128 \times 1$
pool1	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$7 \times 63 \times 63 \times 16$
conv2	$3 \times 3 \times 3$	$1 \times 1 \times 1$	32	$7 \times 31 \times 31 \times 16$
pool2	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$5 \times 29 \times 29 \times 32$
conv3	$3 \times 3 \times 3$	$1 \times 1 \times 1$	64	$5 \times 14 \times 14 \times 32$
pool3	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$3 \times 12 \times 12 \times 64$
conv4	$3 \times 3 \times 3$	$1 \times 1 \times 1$	128	$3 \times 6 \times 6 \times 64$
pool4	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$1 \times 4 \times 4 \times 128$
fc5		$1 \times 1 \times 1$	256	512
fc6		$1 \times 1 \times 1$	64	256

Table 3.3: V2P-Sync audio embedding neural network architecture.

Layer	Support	Stride	Filters	Input
conv1	3×5	1×1	16	$16 \times 40 \times 1$
pool1	1×2	1×2	$14 \times 36 \times 16$	
conv2	3×4	1×1	32	$14 \times 36 \times 16$
conv3	3×4	1×1	32	$12 \times 15 \times 32$
pool3	1×2	1×2	$10 \times 12 \times 32$	
conv4	3×3	1×1	64	$10 \times 6 \times 32$
conv5	3×3	1×1	64	$8 \times 4 \times 64$
conv6	3×2	1×1	128	$6 \times 2 \times 64$
fc7		1×1	256	512
fc8		1×1	64	256

3.5 An efficient spatiotemporal model of visual speech recognition

This work introduces the V2P model, which consists first of a *3d convolutional module* for extracting spatiotemporal features from a given video clip. These features are then aggregated over time with a *temporal module* which outputs a sequence of phoneme distributions. Given input video clips and target phoneme sequences as described in the previous section, the model is trained using the *CTC* loss function. Finally, at test-time, a *decoder* based on finite-state transducers (FSTs) is used to produce a word sequence given a sequence of phoneme distributions. For further architectural details we refer the reader to Table 3.4.

3.5.1 Neural network architecture

Although the use of optical-flow filters as inputs is commonplace in lipreading [138–143], in this work we designed a vision module based on VGG [144] to explicitly address motion feature extraction. We adapted VGG to make it volumetric, which proved crucial in our preliminary empirical evaluation and has been established in previous literature [3]. The intuition behind this is the importance of spatiotemporal relationships in human visual speech recognition, e.g. measuring how lip shape changes over time. Furthermore, the receptive field of the vision module is 11 video frames, roughly 0.36–0.44 seconds, or around twice the typical duration of a phoneme.

One of the main challenges in training a large vision module is finding an effective balance between performance and the imposed constraints of GPU memory. Our vision module consists of 5 convolutional layers with $[64, 128, 256, 512, 512]$ filters. By profiling a number of alternative architectures, we found that high memory usage typically came from the first two convolutional layers. To reduce the memory footprint we limit the number of convolutional filters in these layers, and since the frame is centred around the lips, we omit spatial padding. Since phoneme sequences can be quite long, but with relatively low frame rate (approximately 25–30 fps),

we maintain padding in the temporal dimension and always convolve with unit stride in order to avoid limiting the number of output tokens. Despite tuning the model to reduce the number of activations, we are still only able to fit 2 batch elements on a GPU. Hence, we distribute training across 64 workers in order to achieve a batch size of 128. Due to communication costs, batch normalisation is expensive if one wants to aggregate the statistics across all workers, and using only two examples per batch results in noisy normalisation statistics. Thus, instead of batch normalisation, we use group normalisation [145], which divides the channels into groups and computes the statistics within these groups. This provides more stable learning regardless of batch size.

The outputs of the convolutional stack are then fed into a temporal module which performs longer-scale aggregation of the extracted features over time. In constructing this component we evaluated a number of recurrent neural network and dilated convolutional architectures, the latter of which are evaluated later as baselines. The best architecture presented performs temporal aggregation using a stack of 3 bidirectional LSTMs [74] with a hidden state of 768, interleaved with group normalisation. The output of these LSTM layers is then fed through a final MLP layer to produce a sequence of exactly T conditionally independent phoneme distributions $p(u_t|\mathbf{x})$. This entire model is then trained using the CTC loss we describe next.

This model architecture is similar to that of LipNet [3], but differs in a number of crucial ways. In comparison, LipNet used GRU units and dropout, both of which we found to perform poorly in preliminary experiments. Our model is also much bigger: LipNet consists of only 3 convolutional layers of $[32, 64, 96]$ filters and 3 GRU layers with hidden state of size 256. Although the small size of LipNet means that it does not require any distributed computation to reach effective batch sizes, we will see that this drop in size coincides with a similar drop in performance. Finally, while both models use a CTC loss for training, the architecture used in V2P is trained to predict phonemes rather than characters; as we argue shortly this provides V2P with a much simpler mechanism for representing word uncertainty.

Table 3.4: V2P architecture details.

Layer	Filter size	Stride	Output channels	Input
conv1	$3 \times 3 \times 3$	$1 \times 2 \times 2$	64	$\mathrm{T} \times 128 \times 128 \times 3$
pool1	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$\mathrm{T} \times 63 \times 63 \times 64$
conv2	$3 \times 3 \times 3$	$1 \times 1 \times 1$	128	$\mathrm{T} \times 31 \times 31 \times 64$
pool2	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$\mathrm{T} \times 29 \times 29 \times 128$
conv3	$3 \times 3 \times 3$	$1 \times 1 \times 1$	256	$\mathrm{T} \times 14 \times 14 \times 128$
pool3	$1 \times 2 \times 2$	$1 \times 2 \times 2$		$\mathrm{T} \times 12 \times 12 \times 256$
conv4	$3 \times 3 \times 3$	$1 \times 1 \times 1$	512	$\mathrm{T} \times 6 \times 6 \times 256$
conv5	$3 \times 3 \times 3$	$1 \times 1 \times 1$	512	$\mathrm{T} \times 4 \times 4 \times 512$
pool5	$1 \times 2 \times 2$	$1 \times 1 \times 1$		$\mathrm{T} \times 2 \times 2 \times 512$
bilstm6			768×2	$\mathrm{T} \times 512$
bilstm7			768×2	$\mathrm{T} \times 1536$
bilstm8			768×2	$\mathrm{T} \times 1536$
fc9			768	$\mathrm{T} \times 1536$
fc10			$41 + 1$	$\mathrm{T} \times 768$

3.5.2 Connectionist temporal classification

Connectionist temporal classification (CTC) is a loss function for the parameterisation of distributions over sequences of label tokens, without requiring alignments of the input sequence to the label tokens [146]. Classical approaches to ASR treat the alignment of a sequence of label tokens (e.g. phonemes) with the audio signal as a latent variable, inferring the alignment by an expectation-maximisation-like procedure that alternates between finding alignments using the current model and maximising likelihood. CTC allows one to straightforwardly train discriminative end-to-end models since the marginalisation of the alignment happens in the loss function itself. Hence, it is ubiquitous in modern audio speech recognition (ASR) systems. To see how CTC works, let V denote the set of single-time-step label tokens. To align a label sequence with size-T sequences given by the temporal module, CTC allows the model to output blank symbols $\sqcup$ and repeat consecutive symbols. Let the function $\mathcal{B} : (V \cup \{\sqcup\})^* \to V^*$ be defined such that, given a string potentially containing blank tokens, it deletes adjacent duplicate characters and

removes any blanks. The probability of observing label sequence y can then be obtained by marginalising over all possible alignments of this label,

$$p(y|\mathbf{x}) = \sum_{u \in \mathcal{B}^{-1}(y)} p(u_1|\mathbf{x}) \cdots p(u_T|\mathbf{x}),$$

where $\mathbf{x}$ is input video. For example, if $T = 5$ the probability of sequence 'bee' is given by $p(be␣e␣) + p(␣be␣e) + \cdots + p(bbe␣e) + p(be␣ee)$. Note that there must be a blank between the 'e' characters in order for to avoid collapsing the sequence to 'be'.

Since CTC prevents us from using autoregressive connections to handle inter-time-step dependencies of the label sequence, the marginal distributions produced at each time-step of the temporal module are conditionally independent, as pointed out above. Therefore, to restore temporal dependency of the labels at test-time, CTC models are typically decoded with a beam search procedure that combines the probabilities with that of a language model.

3.5.3 Rationale for phonemes and CTC

In speech recognition, whether on audio or visual signals, there are two main sources of uncertainty: uncertainty in the sounds that are in the input, and uncertainty in the words that correspond to these sounds. This suggests modelling

$$p(\text{words}|\mathbf{x}) = \sum_{\text{phonemes}} p(\text{words}|\text{phonemes}) p(\text{phonemes}|\mathbf{x}) \tag{3.1}$$

$$\approx p(\text{words}|\text{phonemes}) p(\text{phonemes}|\mathbf{x}), \tag{3.2}$$

where the approximation is by the assumption that a given word sequence often has a single or dominant pronunciation. While previous work uses CTC to model characters given audio or visual input directly [3, 147], we argue this is problematic as the conditional independence of CTC time-steps means that the temporal module must assign a high probability to a single sequence in order to not produce spurious modes in the CTC distribution.

To explain why modelling characters with CTC is problematic, consider two character sequences "fare" and "fair" with the same pronunciation (i.e. /fɛ:/).

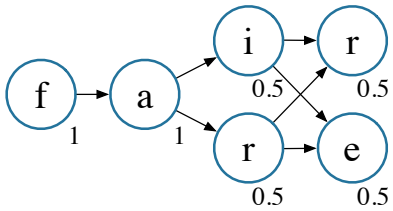


Figure 3.4: Example illustrating the issues with modelling characters with CTC.

The difficulty we will describe is independent of the model used, so we will consider a simple unconditional model where each character c is assigned probability given by the parameters $\pi_t^c = P(u_t = c)$ and the probability of a sequence is given by its product, e.g. $p(\text{fare}) = \pi_1^{\text{f}}\pi_2^{\text{a}}\pi_3^{\text{r}}\pi_4^{\text{e}}$. The maximum likelihood estimate, $\arg\max_\pi p(\text{fare})p(\text{fair})$, however, assigns equal 1/4 probability to each of "fare", "fair", "faie", "farr", as shown in Figure 3.4, resulting in two undesirable words. Ultimately this difficulty arises due to the independence assumption of CTC and the many-to-many mapping of characters to words[2]. This same difficulty arises if we replace the parameters above with the outputs of a network mapping from videos to tokens. Using phonemes, which have a one-to-many map to words, allows the temporal model to only model sound uncertainty, and the word uncertainty can instead be handled by the decoder described below.

Alternatively to using phonemes with CTC, some previous work solves this problem using RNN transducers [148] or sequence-to-sequence with attention [79], which jointly model all sources of uncertainty. However, Prabhavalkar et al. [149] showed in the context of acoustic speech recognition that these models were unable to significantly outperform a baseline CTC model (albeit using context-dependent phonemes and further sequence-discriminative training) when combined with a decoding pipeline similar to ours. Hence, for reasons of performance and easier model training, especially important with our large model, we choose to output phonemes rather than words or characters directly. Additionally, and crucial for

[2]Languages such as Korean, where there is a one-to-one correspondence between pronunciation and orthography, do not give rise to such discrepancies.

many applications, CTC also provides extra flexibility over alternatives. The fact that the lexicon (phoneme to word mapping) and language model are separate and part of the decoder, affords one the ability to trivially change the vocabulary and language model (LM) arbitrarily. This allows for visual speech recognition in narrower domains or updating the vocabulary and LM with new words without requiring retraining of the phoneme recognition model. This is nontrivial in other models, where the language model is part of the RNN.

3.5.4 Decoding using finite-state transducers

As described earlier, our model produces a sequence of phoneme distributions; given these distributions we use decoding method using finite-state transducers (FSTs) to arrive at word sequences. An FST is a type of finite-state automaton that maps between two sets of symbols (or strings). In the following paragraphs we will give a brief introduction to FSTs following Mohri et al. [150].

3.5.5 Finite-state acceptors

Finite automata, also know as finite-state acceptors, are used widely in automatic speech recognition [e.g. 151, 152]. More specifically, these automata are defined by a set of states, the initial state or input symbols (including ϵ, corresponding to no input), and a set of final states or output symbols, and a set of transitions, which map between a source state, a destination state, and label. These automata accept a string that can be read along a path from an initial to a final state, and thus as a whole it represents the set of strings that it can accept.

A weighted finite-state acceptor (WFSA), has additional assigned weights for the transitions and final states. Therefore, each accepted string is assinged a weight, which is computed as the accumulated weights along the accepting paths including final weights. Thus, a weighted acceptor additionally associates to each

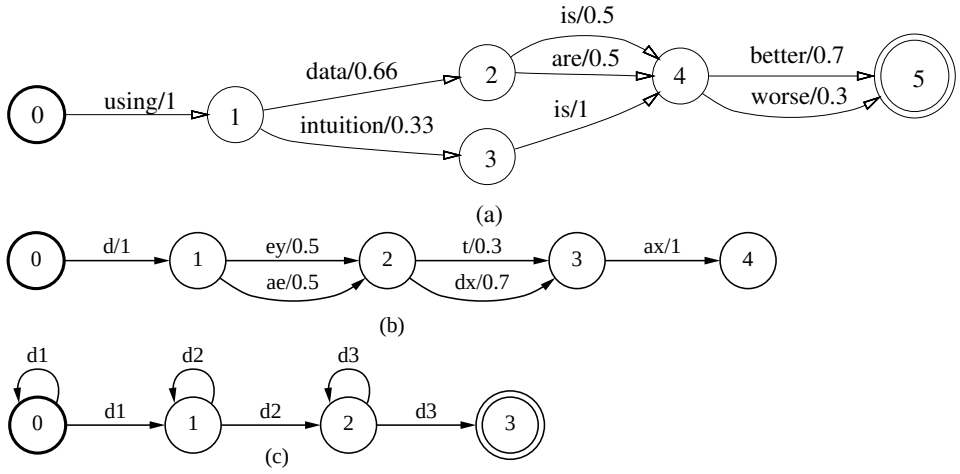


Figure 3.5: Weighted finite-state acceptor examples by Mohri et al. [150]. By convention, the states are represented by circles and marked with their unique number. The initial state is represented by a bold circle, final states by double circles. The label l and weight w of a transition are marked on the corresponding directed arc by l/w.

accepted string the accumulated weights of their accepting paths. Figure 3.5 depicts three examples of WFSAs.

3.5.6 Finite-state transducers

An FST is a finite-state acceptor with the addition of a finite output symbol set (also including ϵ; possibly different from the input symbol set), and transitions also include an output symbol. where state transitions consist of an input symbol, an output symbol. A path through an FST represents an input string that maps to an output string. Since the input and output symbol sets include ϵ, the input and output strings need not be the same length, and an input string has zero or more outputs due to non-determinicity. An input has zero or more outputs, due to nondeterminicity. Similarly to WFSA, weighted FSTs (WFSTs) additionally add a weight to each transition. More generally, the weight operations for a WFST can be specified by a semiring [153–155] that, crucially, allows for union, composition, closure, and other operations. Every transition and final state is assigned a weight; the two operations in the semiring are used to compute the weights of a path

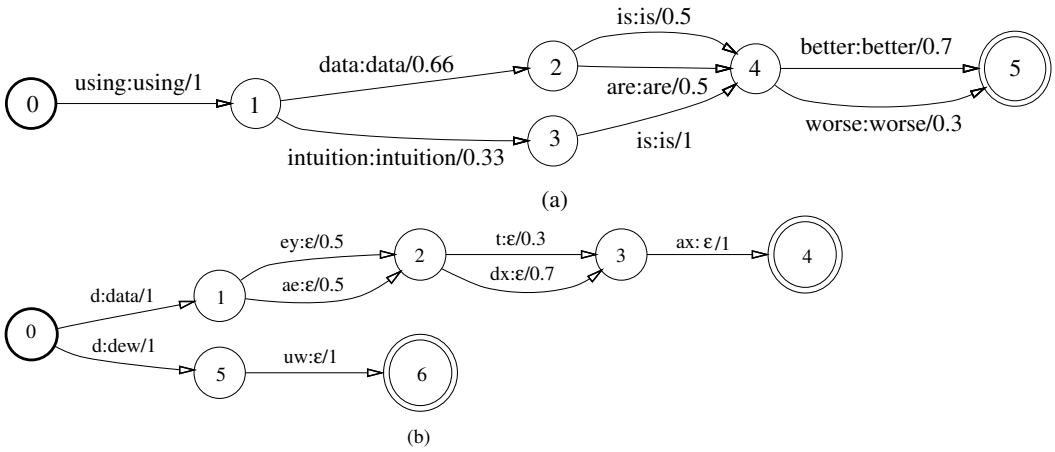


Figure 3.6: Weighted finite-state transducer examples by Mohri et al. [150]. The examples are similar to the WFSA in Figure 3.5 with additional output labels introduced on each transition. The input label i, the output label o, and weight w of a transition are marked on the corresponding directed arc by $i : o/w$.

and perform the aforementioned FST operations. With a probability semiring the operations are $(+,\times)$; in the tropical semiring which can be used to perform Viterbi decoding with negative log probabilities, $(+, \min)$.

In speech recognition, FSTs are used to represent phoneme postprocessing (e.g. when triphones or context-dependent phonemes are used), lexicons (mappings from sequences of phonemes to a word), and language models. In our work we make use of a combination of three individual FSTs. The first *CTC postprocessing FST* removes duplicate symbols and CTC blanks. Next, a *lexicon FST* maps input phonemes to output words. Third, an *n-gram language model with backoff* can be represented as a weighted FST from words to words. In our case, we use a 5-gram language model with Katz smoothing with about 50 million n-grams and a vocabulary size of about one million. The composition of these three FSTs results in another weighted FST that transduces from phoneme sequences to (reweighted) word sequences. Finally, a search procedure is employed to transduce likely words from the phoneme recognition model.

3.6 Experimental evaluation

We examine the performance of V2P trained on LSVSR with hyperparameters tuned on a validation set. We evaluate it on a held-out test set of randomly selected videos from LSVSR, roughly 37 minutes long, containing approximately 63,000 video frames and 7100 words. In order to simulate a real-world setting where a person would be speaking to a camera under controlled conditions, we removed blurry videos from the validation and the test set by thresholding the variance of the Laplacian of each frame [156]. However, blurry videos were kept in the training set as a form of data augmentation. Finally, we describe and compare against a number of alternate methods from previous work. In particular, we show that our system gives significant performance improvements over professional lipreaders as well previous state-of-the-art methods for visual speech recognition.

In each case, the network architecture is optimised using Adam [137] with a learning rate of 10^{-4} and default hyperparameters: first and second momentum coefficients 0.9 and 0.999 respectively, and $\epsilon = 10^{-8}$ for numerical stability. Furthermore, to accelerate learning, a curriculum schedule limits the video duration, starting from 2 seconds and gradually increasing to a maximum length of 12 seconds over 200,000 training steps. Finally, image transformations are also applied to augment the image frames to help improve invariance to filming conditions. This is accomplished by first randomly mirroring the videos horizontally, followed by random changes to brightness, contrast, saturation, and hue.

Professional lipreaders. We consulted a professional lipreading company to measure the difficulty of LSVSR and hence the impact that such a model could have. Since the inherent ambiguity in lipreading necessitates relying on context, we conducted experiments both with and without context. In both cases, we generate modified clips from our test set, but cropping the whole head in the video, as opposed to just the mouth region used by our model. The lipreaders could view the video up to 10 times, at half or normal speed each time. To measure without-context performance, we selected clips with transcripts that had at least 6 words. To measure

how much context helps performance, we selected clips with at least 12 words, and presented to the lipreader the first 6 words, the title, and the category of the video, then asked them to transcribe the rest of the clip. The lipreaders transcribed a subset of our test set containing 153 and 274 videos with and without context, respectively.

Audio-Ph. For comparison and as an approximate bound on performance, we also train an audio speech recognition model on the audio of the utterances, with the architecture based on Deep Speech 2 [147], but trained to predict phonemes rather than characters.

Baseline-LipNet-Ch. Using our training setup, we replicate the character-level CTC architecture of LipNet [3]. As with the phoneme models, we use an FST decoding pipeline and the same language model, but instead of a phoneme-based lexicon we use a character-level one as described in [151].

LipNet-Ph. We train LipNet to predict phonemes and use the same FST-based decoding pipeline and language model.

Baseline-LipNet-Large-Ph. Recall from the earlier discussion that LipNet uses dropout, whereas V2P makes heavy use of group normalization, crucial for our small batches per worker. For a fair size-wise comparison, we introduce a replica of V2P, that uses GRUs, dropout, and no normalization. Preliminary experiments on batch normalization and GRUs were not promising.

Baseline-Seq2seq-Ch. Using our training setup, we compared to a variant of the previous state-of-the-art sequence-to-sequence architecture of WAS that predicts character sequences [79]. Although their implementation was followed as closely as possible, training end-to-end quickly exceeded the memory limitations of modern GPUs. To work around these problems, the authors kept the convolutional weights fixed using a pretrained network from audio-visual synchronisation classification [120], which we were unable to use as their network inputs were processed differently. Instead, we replace the 2D convolutional network with the *improved* lightweight 3D visual processing network of V2P. From our empirical evaluation, including preliminary experiments not reported here and as shown by our earlier work [3], we believe that the 3D spatiotemporal aggregation of features benefits

Method	Param.	PER	WER
Professional w/o context	–	–	92.9 ± 0.9
Professional w/ context	–	–	86.4 ± 1.4
Audio-Ph	58M	12.5 ± 0.5	18.3 ± 0.9
Baseline-LipNet-Ch	7M	–	93.0 ± 0.6
Baseline-LipNet-Ph	7M	65.8 ± 0.4	89.8 ± 0.5
Baseline-Seq2seq-Ch	15M	–	76.8 ± 0.8
Baseline-LipNet-Large-Ph	40M	53.0 ± 0.5	72.7 ± 1.0
V2P-FullyConv	29M	41.3 ± 0.6	51.6 ± 1.2
V2P-NoLM	49M	33.6 ± 0.6	53.6 ± 1.0
V2P	**49M**	$\mathbf{33.6 \pm 0.6}$	$\mathbf{40.9 \pm 1.2}$

Table 3.5: Performance evaluation on LSVSR test set. Columns show phoneme, character, and word error rates, respectively. Standard deviations are bootstrap estimates.

performance. After standard beam search decoding, we use the same 5-gram word LM as used for the CTC models to perform reranking.

V2P-FullyConv. Identical to V2P, except the LSTMs in the temporal aggregation module are replaced with 6 dilated temporal convolution layers with a kernel size of 3 and dilation rates of [1,1,2,4,8,16], yielding a fully convolutional model with 12 layers.

V2P-NoLM. Identical to V2P, except during decoding, where the LM is replaced with a dictionary consisting of 100k words. The words are then weighted by their smoothed frequency in the training data, essentially a uni-gram language model.

3.6.1 Results

Table 3.5 shows the phoneme error rate, character error rate, and word error rate for all of the models, and the number of parameters for each. The error rates are computed as the sum of the edit distances of the predicted and ground-truth sequence pairs divided by total ground-truth length. We also compute and display the standard error associated with each rate, estimated by bootstrap sampling.

These results show that the variant of LipNet tested in this work is approximately able to perform on-par with professional lipreaders with WER of 86.4 and 89.8

respectively, even when the given professional is given additional context. It is worth mentioning, that lipreading professionals had a WER of 73.8 in the LRS dataset as reported by Chung et al. [79]. Similarly, we see that the WAS variant provides a substantial reduction to this error, resulting in a WER of 76.8. However, the full V2P method presented in this work is able to further halve the WER, obtaining a value of 40.9 at testing time. Interestingly, we see that although the bi-directional LSTM provides the best performance, using a fully-convolutional network still results in performance that is significantly better than all previous methods. Finally, although we see that the full V2P model performs best, removing the language model results only in a drop of approximately 13 WER to 53.6.

3.6.2 Phonemic analysis

By predicting phonemes directly, we also side-step the need to design phoneme-to-viseme mappings [116]. The inherent uncertainty is instead modelled directly in the predictive distribution. For instance, using edit distance alignments of the predictions to the ground-truths, we can determine which phonemes were most frequently erroneously confused, included or missed. To compute the confusion matrix in Figure 3.7 and the insertion/deletion chart in Figure 3.8, we first compute the edit distance dynamic programming matrix between each predicted sequence of phonemes and the corresponding ground-truth. Then, a backtrace through this matrix gives an alignment of the two sequences, consisting of edit operations paired with positions in the prediction/ground-truth sequences.

Counting the correct phonemes and the substitutions yields the confusion matrix Figure 3.7. The reader can note that the diagonal is strongly dominant. A few groups are commonly confused as expected due to their visual similarity, such as {/d/, /n/, /t/}, and to a lesser extent {/b/, /p/}. Counting insertions/deletions yields Figure 3.8 in the main text, showing which phonemes are most commonly omitted (deleted), or less frequently, erroneously inserted.

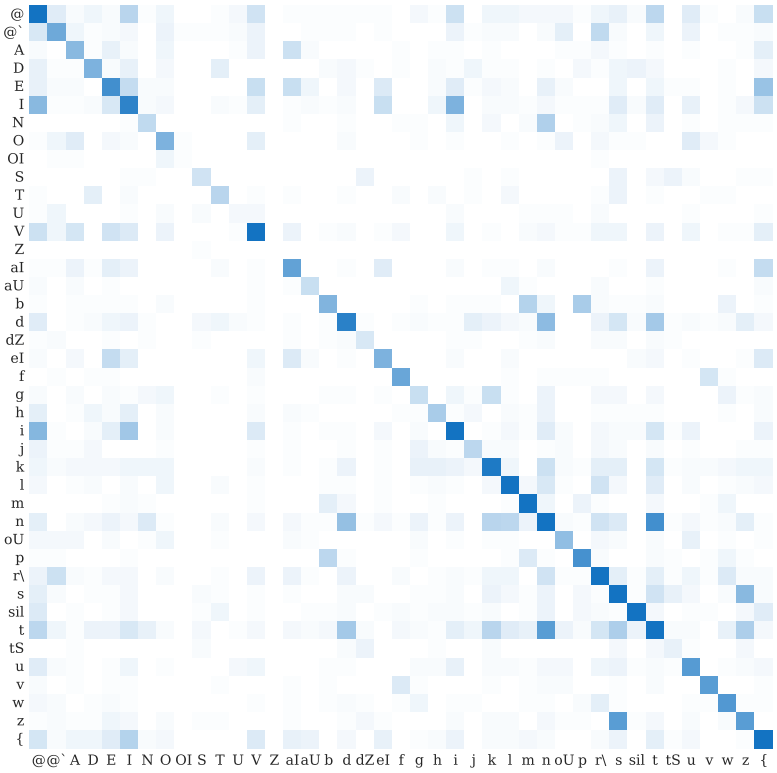


Figure 3.7: Phoneme confusion matrix for V2P, estimated by computing the edit distance alignment between each predicted sequence of phonemes and the corresponding ground-truth, and counting the correct phonemes and the substitutions. The diagonal values are scaled downwards to de-emphasize the correct phonemes. Blue indicates more substitutions occurred.

Figure 3.8: This heatmap shows which insertion and deletion errors were most common on the test set. Blue indicates more insertions or deletions occurred.

Here we normalise the rates of deletions vs insertions, however empirically we saw that deletions were much more common than inclusions. Among these errors, the most common include phonemes that are often occluded by the teeth (/d/, /n/, and /t/) as well as the most common English vowel /@/.

We further study the predictive performance of V2P under different head rotations of which the model was trained. Figure 3.9 shows the WER at all pan and tilt angles in $[-30°, 30°]$. As it can be seen, V2P performs similarly at all pan and title angles.

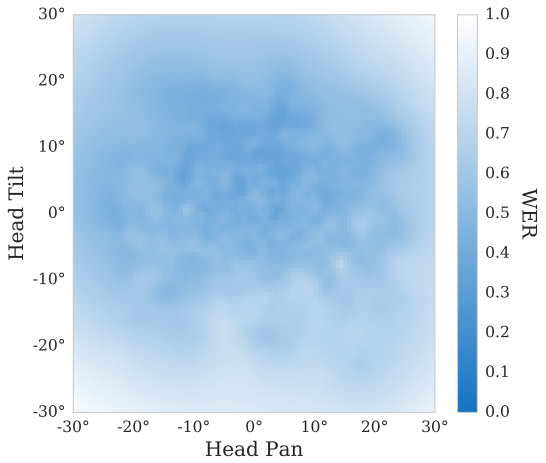


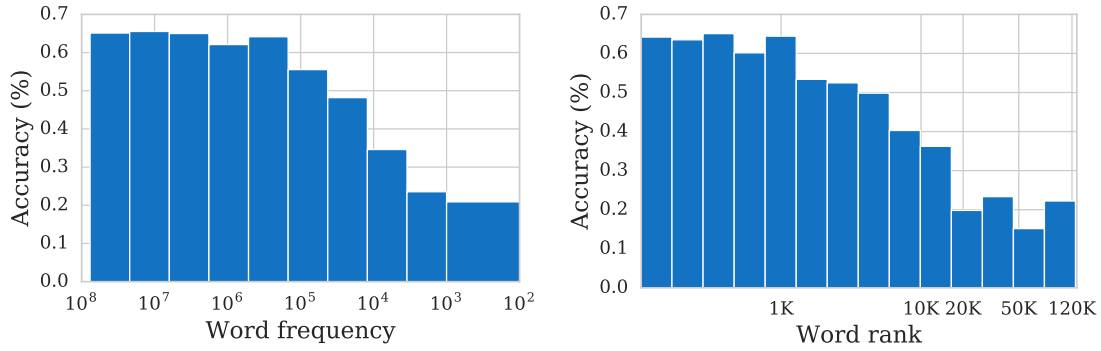
Figure 3.9: Heatmap showing the performance of V2P on different head rotations. Tilt and pan axes are in degrees. As shown, it performs similarly at all pan and tilt angles in $[-30°, 30°]$, the range at which it was trained.

Finally, by differentiating the likelihood of the phoneme sequence with respect to the inputs using guided backpropagation [157], we compute the saliency maps shown in Figure 3.11 as a white overlay. The entropy at each time-step of the phoneme predictive distribution is shown as well.

3.6.3 Vocabulary analysis

In this section we study the statistical performance of V2P in the large vocabulary of LSVSR. As aforementioned, LSVSR has a vocabulary of 127,055 words, which is an order of magnitude larger than previous works (17,428 words [79]). Our goal is to analyse the impact of the less frequent words in the performance of V2P.

By generating the edit distance alignments of the predictions to the ground-truth words, we use the insertion and substitution counts to approximate the per-word accuracy of V2P in our held-out test set. More specifically, in Figure 3.10 we plot the accuracy against the word frequency and the work rank. This analysis is an approximation as the phoneme to word mapping is handled by the FST decoder which incorporates an external language model trained over a much larger portion of YouTube transcripts. The accuracy of V2P is about 65% for words that appear 10^8 to



(a) Accuracy vs word frequency. **(b)** Accuracy vs word rank.

Figure 3.10: Using the edit distance alignments of the predictions to the ground-truth words, we use the insertion and substitution counts to approximate the per-word accuracy of V2P in the LSVSR held-out test set. The accuracy of V2P is about 65% for words that appear 10^8 to 10^5 times in the training set, and gradually drops to 20% for words seen only 100 times or words with rank larger than 20,000.

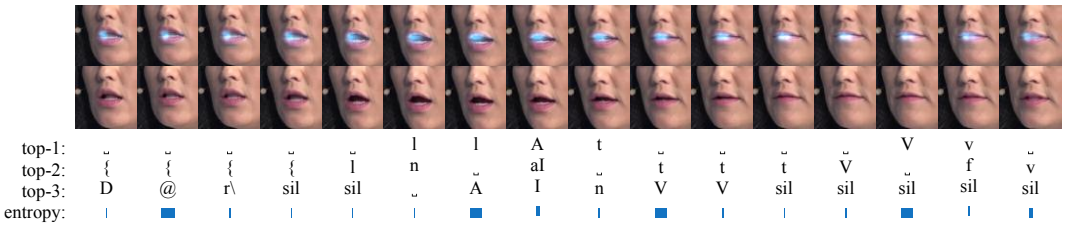
10^5 times in the training set. The performance gradually drops to 20% for words seen only 100 times or words with rank larger than 20,000. It is interesting to note that even in these rare cases, and in a dataset with a vocabulary an order of magnitude larger than previous literature, the performance of V2P only drops to a third.

3.6.4 Generalisation

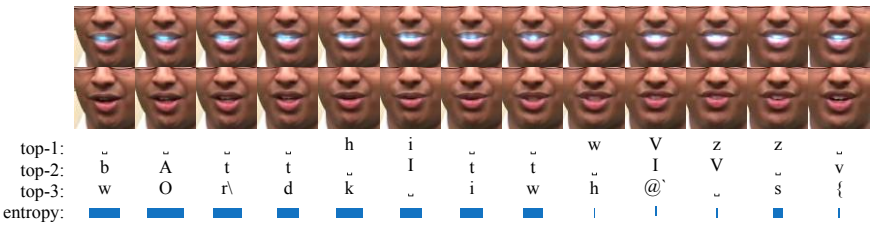
Contemporaneously with our work, Afouras et al. [158] presented LRS3-TED, a dataset generated from English language talks available online. Using pre-learned features, Afouras et al. [159] presented sequence-to-sequence and CTC, character-level self-attention transformer models achieving a WER of 57.9% and 61.8% respectively. To demonstrate the generalization power of V2P, we evaluate on LRS3-TED [158] and compare it to the TM-seq2seq model of [159]. Despite the fact that we do not train or fine-tune V2P on LRS3-TED, our approach still outperforms the state-of-the-art model trained on that dataset. Unlike LSVSR, LRS3-TED includes faces at angles between $\pm 90°$ instead of $\pm 30°$, and clips may be shorter than one second. Thus, we conducted two experiments. First, we evaluated performance on a subset of the LRS3-TED test set filtered according to

the same protocol used to construct LSVSR, by removing instances with larger face angles and shorter clips, and second, on the full unfiltered test set. Despite the fact that we do not train or fine-tune V2P on LRS3-TED, V2P achieves WERs of 47.0 ± 1.6 and 55.1 ± 0.9 respectively, outperforming TM-seq2seq's 57.9. This shows that V2P is able to generalise well, achieving state-of-the-art performance on datasets with different conditions on which it was not trained. Due to the difficulty of obtaining a continually front-on view of a face at a sufficiently high resolution without an individual's consent, the model is not suited for lipreading in scenarios such as surveillance.

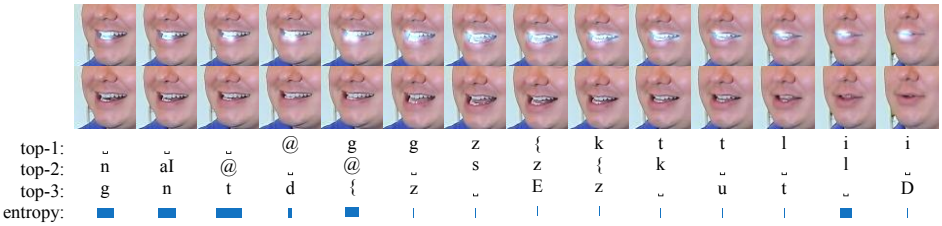
(a) Transcript: “sharing” - ground truth phonemes: /S E r\@ N/.



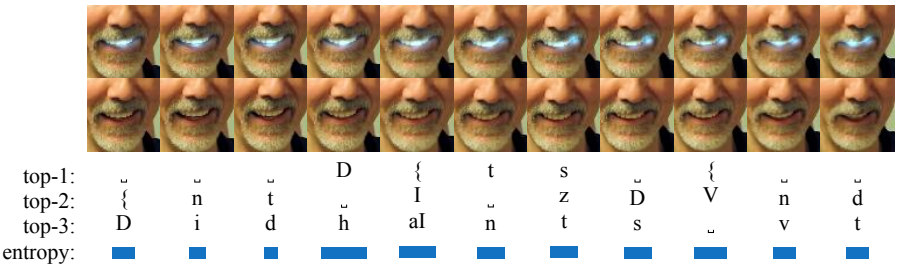
(b) Transcript: “lot of”, ground truth phonemes: /l A t V v/.



(c) Transcript: “how was”, ground truth phonemes /I t w V z/.



(d) Transcript: “exactly”, ground truth phonemes: /@ g z { k t l i/.



(e) Transcript: “that’s a”, ground truth phonemes: /D { t s {/.

Figure 3.11: Saliency maps, the top-3 predictions of each frame and the ground truth phonemes.

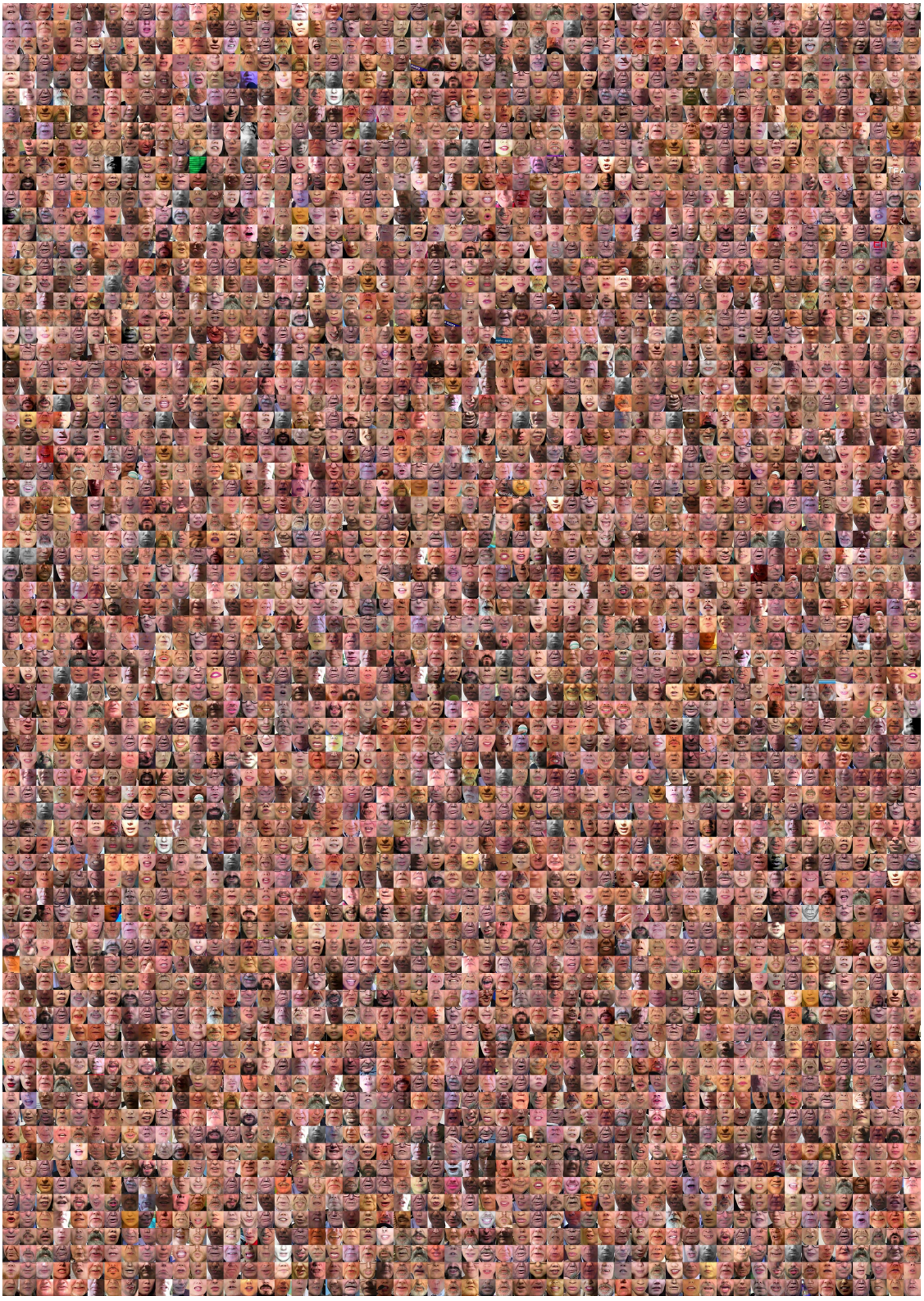


Figure 3.12: Random sample of test-set lip images from LSVSR. This illustrates the substantial diversity in our dataset.

3.7 Conclusion

We presented two novel verbal communication recognition lipreading models, LipNet and V2P. LipNet, was the first end-to-end sentence-level lipreading model, mapping sequences of lip frames to characters. Furthermore, to facilitate further research in the field we made the LipNet source code available online: `https://github.com/bshillingford/LipNet`. The follow-up work on V2P presented a large-scale model for producing phoneme and word sequences from processed lip frames. To train V2P we designed a data processing pipeline used to construct a vast dataset—an order of magnitude greater than all previous approaches both in terms of vocabulary and the sheer number of example sequences, on which V2P was capable of nearly halving the error rate of the previous state-of-the-art methods. The combination of methods in this work represents a significant improvement in lipreading performance, a technology which can enhance automatic speech recognition systems, and which has enormous potential to improve the lives of speech impaired patients worldwide.

Finally, in this chapter we have shown the effectiveness of end-to-end learning for the recognition of visual verbal communication. Furthermore, we conducted a preliminary analysis and showed the potential of the methods presented in medical applications for people with speech impairments. As a next step we will study the synthesis of communication messages and more specifically in the case of verbal communication.

4 Synthesis

Having presented novel methods to improve the recognition of verbal communication, we focus on the synthesis of verbal communication from a given text. However, such methods require an enormous amount of speech data. In this chapter, we investigate whether meta-learning can be used to improve the sample efficiency of these methods allowing them to be used in limited data medical settings. More specifically, we present SEA a meta-learning approach for adaptive text-to-speech (TTS) with few data. During training, a multi-speaker model is learnt using a shared conditional WaveNet core and independent learned embeddings for each speaker. The aim of training is not to produce a neural network with fixed weights, which is then deployed as a TTS system. Instead, the aim is to produce a network that requires few data at deployment time to rapidly adapt to new speakers. Furthermore, the following three strategies are evaluated: (a) learning the speaker embedding while keeping the WaveNet core fixed, (b) fine-tuning the entire architecture with stochastic gradient descent, and (c) predicting the speaker embedding with a trained neural network encoder. Our experimental evaluation shows that these approaches are successful at adapting the multi-speaker neural network to new speakers, obtaining state-of-the-art results in voice similarity with merely a few minutes of audio data from new speakers.

4.1 Introduction

The process of communication between two participants consists of the following entities: a meaning, a message, and a signal [16]. First an abstract meaning is converted to a message, this process is called message *generation*. Afterwards, this message gets *encoded* to a signal which gets transmitted to the other participant. Which does the reverse process to *decode* and *understand* it. This chapter focuses on *encoding*. In the case of writing, it is writing encoding, while for speech it is speech encoding. Speech encoding by computer is more commonly known as *speech synthesis*. Speech synthesis from a given text is called *text-to-speech synthesis* (TTS), which is the topic of this chapter.

In recent years, text-to-speech synthesis has become widespread in the form of consumer products such as virtual personal assistants, in-home devices, cars, and a plethora of other applications [160]. Some of the applications of TTS with impact for the greater good, include medical applications to patients with speaking impairments [161–163]. The common factor of all deployed systems, is the need for a large amount of data to train a model, which is common methodology in deep learning. It has been particularly successful in speech recognition [28], machine translation [164] and image recognition [165, 166].

In this work, instead focuses in few-shot meta-learning. Here the objective of training with many data is not to learn a fixed-parameter classifier, but rather to learn a *"prior" neural network*. This prior TTS network can be adapted rapidly, using few data, to produce TTS systems for new speakers at deployment time. That is, the intention is not to learn a fixed final model, but rather to learn a model prior that harnesses few data at deployment time to learn new behaviours rapidly. The output of training is not longer a fixed model, but rather a fast learner.

Biology provides motivation for this line of research. It may be argued that evolution is a slow adaptation process that has resulted in biological machines with the ability to adapt rapidly to new data during their lifetimes. These machines are born with strong priors that facilitate rapid learning.

We consider a meta-learning approach where the model has two types of parameters: task-dependent parameters and task-independent parameters. During training, we learn all of these parameters but discard the task-dependent parameters for deployment. The goal is to use few data to learn the task-dependent parameters for new tasks rapidly. Task-dependent parameters play a similar role to latent variables in classical probabilistic graphical models. Intuitively, these variables introduce flexibility, thus making it easier to learn the task-independent parameters. For example, in classical HMMs, knowing the latent variables results in a simple learning problem of estimating the parameters of an exponential-family distribution. In neural networks, this approach also facilitates learning when there is clear data diversity and categorisation, this work demonstrates this for adaptive TTS [167, 16]. In this setting, speakers correspond to tasks. During training there are many speakers, and it is therefore helpful to have task-dependent parameters to capture speaker-specific voice styles. At the same time, it is useful to have a large model with shared parameters to capture the generic process of mapping text to speech. To this end, we employ the WaveNet model.

WaveNet [29] is an autoregressive generative model for audio waveforms that has yielded state-of-art performance in speech synthesis. This model was later modified for real-time speech generation via probability density distillation into a feed-forward model [168]. A fundamental limitation of WaveNet is the need for hours of training data for each speaker. In this work, we describe a new WaveNet training procedure that facilitates adaptation to new speakers, allowing the synthesis of new voices from no more than 10 minutes of data with high sample quality.

We propose several extensions of WaveNet for sample-efficient adaptive TTS. First, we present two non-parametric adaptation methods that involve fine-tuning either the speaker embeddings only or all the model parameters given few data from a new speaker. Second, we present a parametric text-independent approach whereby an auxiliary network is trained to predict new speaker embeddings.

The experiments will show that all the proposed approaches, when provided with just a few seconds or minutes of recording, can generate high-fidelity utterances

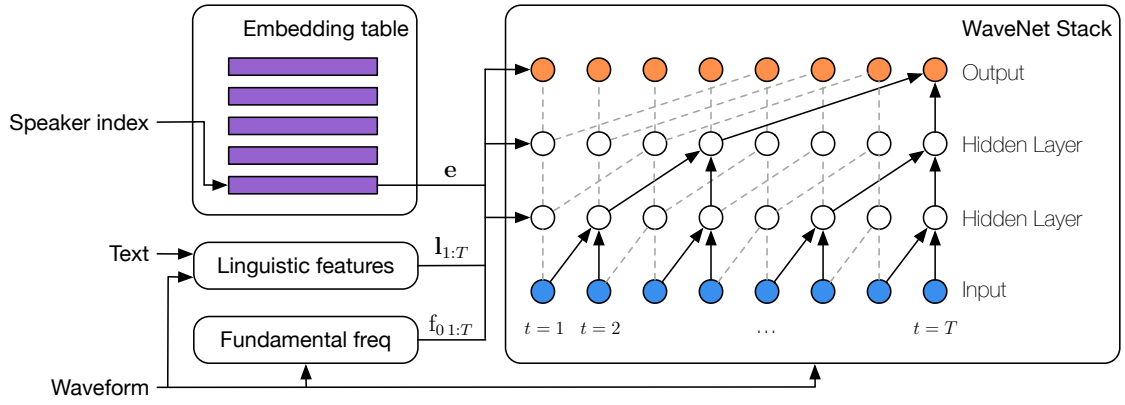


Figure 4.1: Architecture of the WaveNet model for few-shot voice adaptation.

that closely resemble the vocal tract characteristics of a demonstration speaker, particularly when the entire model is fine-tuned end-to-end. When fine-tuning by first estimating the speaker embedding and subsequently fine-tuning the entire model, we achieve state-of-the-art results in voice similarity to target speakers. These results are robust across speech datasets recorded under different conditions and, moreover, we demonstrate that the generated samples are capable of confusing the state-of-the-art text-independent speaker verification system [169].

TTS techniques require hours of high-quality recordings, collected in controlled environments, for each new voice style. Given this high cost, reducing the length of the training dataset could be valuable. For example, it is likely to be very beneficial when attempting to restore the voices of patients who suffer from voice-impairing medical conditions. In these cases, long high quality recordings are scarce.

4.2 WaveNet

4.2.1 Architecture

WaveNet is an autoregressive model that factorises the joint probability distribution of a waveform, $\mathbf{x} = \{x_1, \ldots, x_T\}$, into a product of conditional distributions

using the probabilistic chain rule:

$$p(\mathbf{x}|\mathbf{h};\mathbf{w}) = \prod_{t=1}^{T} p(x_t|\mathbf{x}_{1:t-1},\mathbf{h};\mathbf{w}),$$

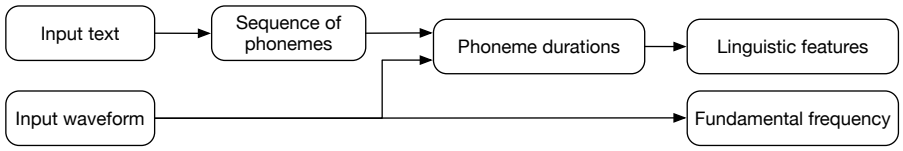
where x_t is the t-th time-step sample, and $\mathbf{h}$ and $\mathbf{w}$ are respectively the conditioning inputs and parameters of the model. To train a multi-speaker WaveNet for text-to-speech synthesis, the conditioning inputs $\mathbf{h}$ consist of the speaker embedding vector $\mathbf{e}_s$ indexed by the speaker identity s, the linguistic features $\mathbf{l}$, and the logarithmic fundamental frequency $\mathbf{f}_0$ values. $\mathbf{l}$ encodes the sequence of phonemes derived from the input text, and $\mathbf{f}_0$ controls the dynamics of the pitch in the generated utterance. Given the speaker identity s for each utterance in the dataset, the model is expressed as:

$$p(\mathbf{x}|\mathbf{l},\mathbf{f}_0;\mathbf{e}_s,\mathbf{w}) = \prod_{t=1}^{T} p(x_t|\mathbf{x}_{1:t-1},\mathbf{l},\mathbf{f}_0;\mathbf{e}_s,\mathbf{w}),$$

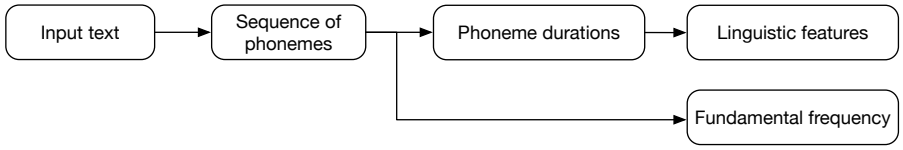
where a table of speaker embedding vectors $\mathbf{e}_s$ (*Embedding* in Figure 4.1) is learned alongside the standard WaveNet parameters. These vectors capture salient voice characteristics across individual speakers, and provide a convenient mechanism for generalising WaveNet to the few-shot adaptation setting in this work. The linguistic features $\mathbf{l}$ and fundamental frequency values $\mathbf{f}_0$ are both time-series with a lower sampling frequency than the waveform. Thus, to be used as local conditioning variables they are upsampled by a transposed convolutional network. During training, $\mathbf{l}$ and $\mathbf{f}_0$ are extracted by signal processing methods from pairs of training utterance and transcript, and during testing, those values are predicted from text by existing models [170], with more details in Section 4.2.2.

4.2.2 Linguistic features and fundamental frequency

The same pipeline as Van Den Oord et al. [29] was employed to generate the linguistic features $\mathbf{l}$ and fundamental frequency $\mathbf{f}_0$ as illustrated in Figure 4.2. Further details for each step are given in the following paragraphs.



(a) Training and adaptation phase



(b) Inference phase

Figure 4.2: Pipelines to generate linguistic features and fundamental frequency from input text and waveform.

Sequence of phonemes

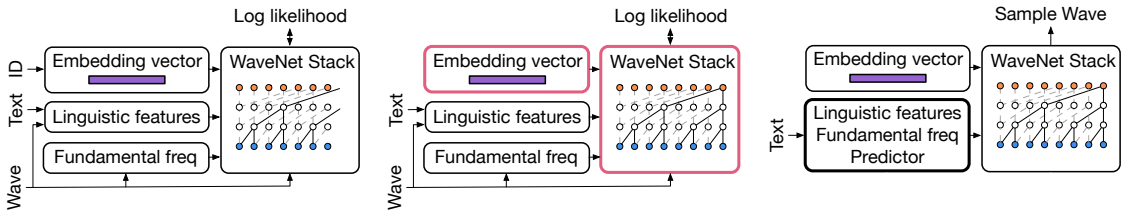
The sequence of phonemes were predicted by the text-analysis front-end pipeline [171] using a tokeniser, POS tagger, dependency parser, text normaliser, lexicon, and grapheme-to-phoneme rules. The contexts associated with the phonemes include syllable, word, phrase, and sentence-level information, such as POS, syllable stress, dependency parser output, and sentence type.

Phoneme duration and fundamental frequency ($\mathbf{f}_0$)

In the training and adaptation phases phoneme durations and the fundamental frequency are extraected from natural speech. $\mathbf{f}_0$ values were extracted using an YIN $\mathbf{f}_0$ tracker [172]. Phone durations were obtained from pairs of text and audio by forced alignment with a flat-start HMM-based aligner. At inference time they are predicted by an LSTM-based $\mathbf{f}_0$ and phone duration predictor [170]. The model was trained with the data of a single speaker, a subset of the WaveNet training data.

Set of linguistic features

Our linguistic features are based on those described in Dall et al. [173, Sec. 2]. The main differences are 1) we do not use ToBI end-tone marking, 2) we use categorical features derived from a dependency parser as those described in Dall et al. [173, Sec. 3].



(a) Training **(b)** Adaptation **(c)** Inference

Figure 4.3: Training (slow, lots of data), adaptation (fast, few data) and inference stages for the SEA-All architecture. The components with bold pink outlines are fine-tuned during the adaptation phase. The purpose of training is to produce a prior. This prior is combined with few data during adaptation to solve a new task. This adapted model is then deployed in the final inference stage.

4.3 Few-shot adaptation with WaveNet

In recent years, a large body of literature uses large datasets to train models to learn an input-output mapping that is then used for inference. In contrast, few-shot meta-learning introduces an additional step, adaptation. In this meta-learning setting, the purpose of training becomes to learn a prior. During adaptation, this prior is combined with few data to rapidly learn a new skill; in this case adapting to a new speakers' voice style. Finally, the new skill is deployed, which in this work we are referring to as inference. These three stages — training, adaptation and inference — are illustrated in Figure 4.3.

We present two multi-speaker WaveNet extensions for few-shot voice adaptation. First, we introduce a non-parametric model fine-tuning approach, which involves adapting either the speaker embeddings or all the model parameters using held-aside demonstration data. Second, and for comparison purposes, we use a parametric approach whereby an auxiliary network is trained to predict the embedding vector of a new speaker using the demonstration data.

4.3.1 Non-parametric few-shot adaptation via fine-tuning

Inspired by few-shot learning we first pre-train a multi-speaker conditional WaveNet model on a large and diverse dataset, as described in Section 4.2. Sub-

sequently, we fine-tune the model parameters by retraining with respect to held-aside adaptation data. Training this WaveNet model to maximise the conditional log-likelihood of the generated audio jointly optimises both the set of speaker parameters $\{\mathbf{e}_s\}$ and the shared WaveNet core parameters $\mathbf{w}$. Next, we extend this method to a new speaker by extracting the $\mathbf{l}$ and $\mathbf{f}_0$ features from their adaptation data and randomly initialising a new embedding vector $\mathbf{e}$. We then optimise $\mathbf{e}$ such that the demonstration waveforms, $\{\mathbf{x}^{(1)}_{\text{demo}}, \ldots, \mathbf{x}^{(n)}_{\text{demo}}\}$, paired with features $\{(\mathbf{l}^{(1)}_{\text{demo}}, \mathbf{f}^{(1)}_{0,\text{demo}}), \ldots, (\mathbf{l}^{(n)}_{\text{demo}}, \mathbf{f}^{(n)}_{0,\text{demo}})\}$, are likely under the model with $\mathbf{w}$ fixed (SEA-EMB[1]):

$$\mathbf{e}_{\text{demo}} = \arg\max_{\mathbf{e}} \sum_i \log p(\mathbf{x}^{(i)}_{\text{demo}} | \mathbf{l}^{(i)}_{\text{demo}}, \mathbf{f}^{(i)}_{0,\text{demo}}; \mathbf{e}, \mathbf{w}).$$

Alternatively, all of the model parameters may be additionally fine-tuned (SEA-ALL):

$$(\mathbf{e}_{\text{demo}}, \mathbf{w}_{\text{finetuned}}) = \arg\max_{\mathbf{e},\mathbf{w}} \sum_i \log p(\mathbf{x}^{(i)}_{\text{demo}} | \mathbf{l}^{(i)}_{\text{demo}}, \mathbf{f}^{(i)}_{0,\text{demo}}; \mathbf{e}, \mathbf{w}).$$

Both methods are non-parametric approaches to few-shot voice adaptation as the number of embedding vectors scales with the number of speakers. However, the training processes are slightly different. Because the SEA-EMB method optimises only a low-dimensional vector, it is far less prone to overfitting, and we are therefore able to retrain the model to convergence even with mere seconds of adaptation data. By contrast, the SEA-ALL has many more parameters that might overfit to the adaptation data. We therefore hold out 10% of our demonstration data for calculating a standard early termination criterion on the log-likelihood of the hold-out data. We also initialise $\mathbf{e}$ with the optimal value from the SEA-EMB method, and we find this initialization significantly improves the generalization performance even with a few seconds of adaptation data.

[1]SEA is short for Sample Efficient Adaptive TTS.

4.3.2 Parametric few-shot adaptation using an embedding encoder

In contrast to the non-parametric approach, whereby a different embedding vector is fitted for each speaker, one can train an auxiliary encoder network to predict an embedding vector for a new speaker given their demonstration data. Specifically, we model:

$$p(\mathbf{x}|\mathbf{l}, \mathbf{f}_0, \mathbf{x}_{\text{demo}}, \mathbf{l}_{\text{demo}}, \mathbf{f}_{0,\text{demo}}; \mathbf{w}) = \prod_{t=1}^{T} p(x_t|\mathbf{x}_{1:t-1}, \mathbf{l}, \mathbf{f}_0; \mathbf{e}(\mathbf{x}_{\text{demo}}, \mathbf{l}_{\text{demo}}, \mathbf{f}_{0,\text{demo}}), \mathbf{w}),$$

where for each training example, we include a randomly selected demonstration utterance from that speaker in addition to the regular conditioning inputs. The full WaveNet model and the encoder network $\mathbf{e}(\cdot)$ are trained together from scratch.

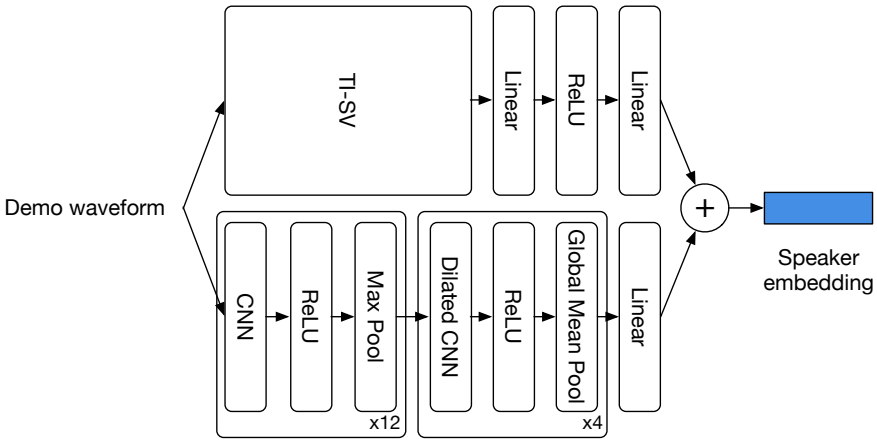


Figure 4.4: Encoder network architecture for predicting speaker embeddings.

Our encoding network is illustrated as the summation of two sub-network outputs in Figure 4.4. The first sub-network is a pre-trained speaker verification model (TI-SV) [169], comprising 3 LSTM layers and a single linear layer. This model maps a waveform sequence of arbitrary length to a fixed 256-dimensional d-vector with a sliding window, and is trained from approximately 36M utterances from 18K speakers extracted from anonymised voice search logs. On top of this we add a shallow MLP to project the output d-vector to the speaker embedding space. The second sub-network comprises 16 1-D convolutional layers. This network reduces

the temporal resolution to 256 ms per frame (for 16 kHz audio), then averages across time and projects into the speaker embedding space. The purpose of this network is to extract residual speaker information present in the demonstration waveforms but not captured by the pre-trained TI-SV model.

Finally, this approach (SEA-Enc) exhibits the advantage of being trained in a transcript-independent setting given only the input waveform, $\mathbf{e}(\mathbf{x}_{\text{demo}})$, and requires negligible computation at adaptation time. However, the learned encoder can also introduce bias when fitting an embedding due to its limited network capacity. As an example, Li et al. [174] demonstrated a typical scenario whereby speaker identity information can be very quickly extracted with deep models from audio signals. Nonetheless, the model is less capable of effectively leveraging additional training than approaches based on statistical methods.

4.3.3 Removing identity-related information

The linguistic features and fundamental frequencies which are used as inputs contain information specific to an individual speaker. As an example, the average voice pitch in the fundamental frequency sequence is highly speaker-dependent. Instead, we would like these features to be as speaker-independent as possible such that identity is modelled via global conditioning on the speaker embedding. To achieve this, we normalise the fundamental frequency values to have zero mean and unit variance separately for each speaker during training, denoted as $\hat{\mathbf{f}}_0 := (\mathbf{f} - \mathbb{E}[\mathbf{f}_s])/\text{std}(\mathbf{f}_s)$. As mentioned earlier, at test time, we use an existing model [170] to predict $(\mathbf{l}, \hat{\mathbf{f}}_0)$.

4.4 Related work

Few-shot learning to build models, where one can rapidly learn using only a small amount of available data, is one of the most important open challenges in machine learning. Recent studies have attempted to address the problem of few-shot

learning by using deep neural networks, and they have shown promising results on classification tasks in vision [175, 176], language [177] and speech recognition [178]. Few-shot learning can also be leveraged in reinforcement learning, such as by imitating human Atari gameplay from a single recorded action sequence [179] or online video [180].

Meta-learning offers a sound framework for addressing few-shot learning. Here, an expensive learning process results in machines with the ability to learn rapidly from few data. Meta-learning has a long history [181, 182], and recent studies include efforts to learn optimization processes [183, 184] that have been shown to extend naturally to the few-shot setting [185]. An alternative approach is model-agnostic meta learning (MAML) [186], which differs by using a fixed optimiser and learning a set of base parameters that can be adapted to minimise any task loss by few steps of gradient descent. This method has shown promise in robotics [187, 188].

In generative modelling, few-shot learning has been addressed from several perspectives, including matching networks [189] and variable inference for memory addressing [190]. Rezende et al. [191] developed a sequential generative model that extended the Deep Recurrent Attention Writer (DRAW) model [58], and Reed et al. [192] extended PixelCNN [193] with neural attention for few-shot autoregressive density modelling. Veness et al. [194] presented a gated linear model able to model complex densities from a single pass of a limited dataset. Early attempts of few-shot adaptation involved the attention models of Reed et al. [192] and MAML [186], but we found both of these strategies failed to learn informative speaker embedding in our preliminary experiments.

There is growing interest in developing neural TTS models that can be trained end-to-end without the need for hand-crafted representations. In this study we focus on extending the autoregressive WaveNet model [29, 168] to the few-shot learning setting to adapt to speakers that were not presented at training time. Other recent neural TTS models include Tacotron 2 [195] (building on [196]) which uses WaveNet as a vocoder to invert mel-spectrograms generated by an attentive sequence-to-sequence model. DeepVoice 2 [197] (building on [198]) introduced a multi-speaker

variation of Tacotron that learns a low-dimensional embedding for each speaker, which was further extended in DeepVoice 3 [199] to a 2,400 multi-speaker scenario. Unlike WaveNet and DeepVoice, the Char2Wav [200] and VoiceLoop [201] models produce World Vocoder Features [202] instead of generating raw audio signals.

Although many of these systems have produced high-quality samples for speakers present in the training set, generalising to new speakers given only a few seconds of audio remains a challenge. There have been several concurrent works to address this few-shot learning problem. The VoiceLoop model introduced a novel memory-based architecture that was extended by Nachmani et al. [203] to few-shot voice style adaptation, by introducing an auxiliary fitting network that predicts the embedding of a new speaker. Jia et al. [204] extended the Tacotron model for one-shot speaker adaptation by conditioning on a speaker embedding vector extracted from a pretrained speaker identity model of Wan et al. [169]. The most similar approached to our work was proposed by Arik et al. [205] for the DeepVoice 3 model. They considered both predicting the embedding with an encoding network and fitting the embedding based on a small amount of adaptation data, but the adaptation was applied to a prediction model for mel-spectrograms with a fixed vocoder.

4.5 Experimental evaluation

In this section, we evaluate the quality of samples of SEA-All, SEA-Emb and SEA-Enc. We first measure the naturalness of the generated utterances using the standard Mean Opinion Score (MOS) procedure. Then, we evaluate the similarity of generated and real samples using the subjective MOS test and objectively using a speaker verification system [169]. Finally, we study these results varying the size of the adaptation dataset.

4.5.1 Experimental setup

We train a WaveNet model for each of our three methods using the same dataset, which combines the high-quality LibriSpeech audiobook corpus [206] and a proprietary speech corpus. The LibriSpeech dataset consists of 2302 speakers from the train speaker subsets and approximately 500 hours of utterances, sampled at a frequency of 16 kHz. The proprietary speech corpus consists of 10 American English speakers and approximately 300 hours of utterances, and we down-sample the recording frequency to 16 kHz to match LibriSpeech. The multi-speaker WaveNet model has the same architecture as [29] except that we use a 200-dimensional speaker embedding space to model the large diversity of voices.

Our few-shot model performance is evaluated using two hold-out datasets. First, the LibriSpeech test corpus consists of 39 speakers, with an average of approximately 52 utterances and 5 minutes of audio per speaker. For every test speaker, we randomly split their demonstration utterances into an adaptation set for adapting our WaveNet models and a test set for evaluation. The subset of utterances used for early termination in Section 4.3.1 is chosen from the adaptation set. There are about 4.2 utterances on average per speaker in the test set and the rest in the adaptation set. Second, we consider a subset of the CSTR VCTK corpus [207] consisting of 21 American English speakers, with approximately 368 utterances and 12 minutes of audio per speaker. We also apply the adaptation/test split with 10 utterances per speaker for test. We emphasise that no data from VCTK was presented to the model at training time. Since our underlying WaveNet model was trained on data largely from LibriSpeech (which was recorded under noisier conditions than VCTK), one might expect that the generated samples on the VCTK dataset contain characteristic artefactss that make generated samples easier to distinguish from real utterances. However, our evaluation using VCTK indicates that our model generalises effectively and that such artefactss are not detectable. Synthetic utterances are provided on our demo webpage[2].

[2] `https://sample-efficient-adaptive-tts.github.io/demo`

Dataset	LibriSpeech		VCTK	
Real utterance	4.38 ± 0.04		4.45 ± 0.04	
Van Den Oord et al. [29]	4.21 ± 0.081			
Nachmani et al. [203]	2.53 ± 1.11		3.66 ± 0.84	
Arik et al. [205]				
adapt embedding	-		2.67 ± 0.10	
adapt whole-model	-		3.16 ± 0.09	
encode & fine-tune	-		2.99 ± 0.12	
Jia et al. [204]				
train on LibriSpeech	$\mathbf{4.12 \pm 0.05}$		$\mathbf{4.01 \pm 0.06}$	
Adaptation data size	*10 sec*	*<5 min*	*10 sec*	*<10 min*
SEA-All (ours)	3.94 ± 0.08	$\mathbf{4.13 \pm 0.06}$	$\mathbf{3.92 \pm 0.07}$	$\mathbf{3.92 \pm 0.07}$
SEA-Emb (ours)	3.86 ± 0.07	3.95 ± 0.07	3.81 ± 0.07	3.82 ± 0.07
SEA-Enc (ours)	3.61 ± 0.06	3.56 ± 0.06	3.65 ± 0.06	3.58 ± 0.06

Table 4.1: Naturalness of the adapted voices using a 5-scale MOS score (higher is better) with 95% confidence interval on the LibriSpeech and VCTK held-out adaptation datasets. Numbers in bold are the best few-shot learning results on each dataset without statistically significant difference. Van Den Oord et al. [29] was trained with 24-hour production quality data, Nachmani et al. [203] used all samples of each new speaker, Arik et al. [205] used 10 samples where each sample is usually up to few tens of seconds in length, and Jia et al. [204] used 5 seconds.

It is worth mentioning, that SEA-Enc requires no adaptation time. Where for SEA-Emb, it takes $5 \sim 10k$ optimising steps to fit the embedding vector, and an additional $100 \sim 200$ steps to fine-tune the entire model using early stopping for SEA-All.

4.5.2 Naturalness of the generated samples (MOS)

We measure the quality of the generated samples by conducting a MOS test, whereby subjects are asked to rate the naturalness of generated utterances on a five-point Likert Scale (1: Bad, 2: Poor, 3: Fair, 4: Good, 5: Excellent). Furthermore, we compare with other published few-shot TTS systems systems, that were developed in parallel to this work. However, the literature uses varying combinations of training data and evaluation splits making comparison difficult. The results presented are from the closest experimental setups to ours.

Table 4.1 presents MOS for the adaptation models compared to real utterances. Two different adaptation dataset sizes are considered; $T = 10$ seconds, and $T \leq 5$ minutes for LibriSpeech ($T \leq 10$ minutes for VCTK). For reference on 16 kHz data, WaveNet trained on a 24-hour production quality speech dataset [29] achieves a score of 4.21, while for LibriSpeech our best few-shot model attains an MOS score of 4.13 using only 5 minutes of data given a pre-trained multi-speaker model. We note that both fine-tuning models produce overall "good" samples for both the LibriSpeech and VCTK test sets, with SEA-All outperforming SEA-Emb in all cases. SEA-All is on par with the state-of-the-art performance on both datasets. The addition of extra adaptation data beyond 10 seconds of audio helps performance on LibriSpeech but not VCTK, and the gap between our best model and the real utterance is also wider on VCTK, possibly due to the different recording conditions.

4.5.3 Voice similarity (MOS)

Beside naturalness, we also measure the similarity of the generated and real voices. The quality of similarity is the main evaluation metric for the voice adaptation problem. We first follow the experiment setup of Jia et al. [204] to run a MOS test for a subjective assessment and then use a speaker verification model for objective evaluation in the next section. In every trial of this test a subject is presented with a pair of utterances consisting of a real utterance and another real or generated utterance from the same speaker, and is asked to rate the similarity in voice identity using a five-scale score (1: Not at all similar, 2: Slightly similar, 3: Moderately similar, 4: Very similar, 5: Extremely similar).

Table 4.2 shows the MOS for real utterances and all the adaptation models under two adaptation data time settings on both datasets. Again, the SEA-All model outperforms the other two models, and the improvement over SEA-Emb scales with the amount of adaptation data. Particularly, the learned voices on the VCTK dataset achieve an average score of 3.97, demonstrating the generalization performance on a different dataset. As a rough comparison, because of varying training setups, the

Dataset	LibriSpeech		VCTK	
Real utterance	4.30 ± 0.08		4.59 ± 0.06	
Jia et al. [204] train on LibriSpeech	3.03 ± 0.09		2.77 ± 0.08	
Adaptation data size	*10 sec*	*<5 min*	*10 sec*	*<10 min*
SEA-All (ours)	$\mathbf{3.41 \pm 0.10}$	$\mathbf{3.75 \pm 0.09}$	$\mathbf{3.51 \pm 0.10}$	$\mathbf{3.97 \pm 0.09}$
SEA-Emb (ours)	3.42 ± 0.10	3.56 ± 0.10	3.07 ± 0.10	3.18 ± 0.10
SEA-Enc (ours)	2.47 ± 0.09	2.59 ± 0.09	2.07 ± 0.08	2.19 ± 0.09

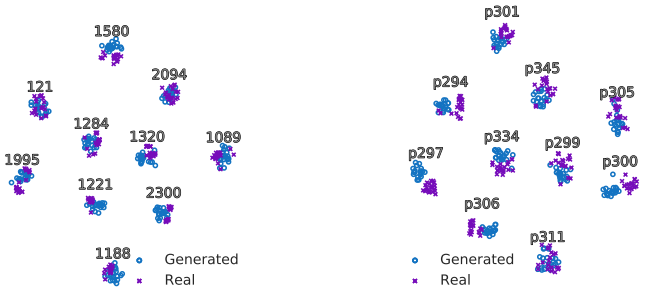
Table 4.2: Voice similarity of generated voices using a 5-scale MOS score (higher is better) with 95% confidence interval on the LibriSpeech and VCTK held-out adaptation datasets.

state of the art system of Jia et al. [204] achieves scores of 3.03 for LibriSpeech and 2.77 for VCTK when trained on LibriSpeech. Their model computes the embedding based on the *d*-vector, similar to our SEA-Enc approach, and performs competitively for the one-shot learning setting, but its performance saturates with 5 seconds of adaptation data. We note the gap of similarity scores between SEA-All and real utterances, which suggests that although the generated samples sound similar to the target speakers, humans can still tell the difference from real utterances.

4.5.4 Voice similarity (speaker verification)

We also apply the state-of-the-art text independent speaker verification (TI-SV) model of [169] to objectively assess whether the generated samples preserve the acoustic features of the speakers. We calculate the TI-SV *d*-vector embeddings for generated and real voices. In Figure 4.5, we visualise the 2-dimensional projection of the *d*-vectors for a SEA-All model trained on $T \leq 5$ minutes of data on the LibriSpeech dataset, and $T \leq 10$ minutes on VCTK. There are clear clusters on both datasets, with a strikingly large inter-cluster distance and low intra-cluster separation. This shows both (1) an ease of correctly identifying the speaker associated with a given generated utterance, and (2) the difficulty in differentiating real from synthetic samples.

A similar figure is presented in [204], but there the generated and real samples do not overlap. This indicates that the method presented in this work generates voices that are more indistinguishable from real ones, when measured with the same verification system. In the following subsections, we further analyze these results.



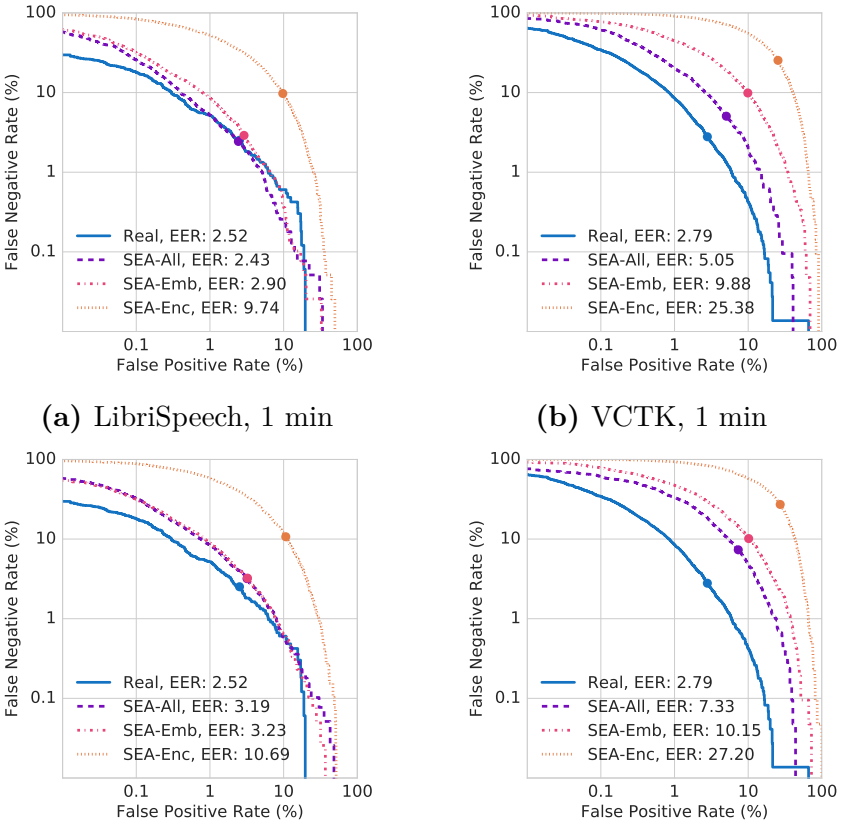
(a) LibriSpeech **(b)** VCTK

Figure 4.5: t-SNE visualization of the d-vector embeddings of real and SEA-ALL-generated utterances, for both the LibriSpeech ($T \leq 5$ mins) and VCTK ($T \leq 10$ mins) evaluation datasets.

Discerning different speakers

We first quantify whether generated utterances are attributed to the correct speaker. Following common practice in speaker verification [169], we select the hold-out test set of real utterances from test speakers as the enrolment set and compute the centroid of the d-vectors for each speaker $\mathbf{c}_i$. We then use the adaptation set of test speakers as the verification set. For every verification utterance, we compute the cosine similarity between its d-vector $\mathbf{v}$ and a *randomly* chosen centroid $\mathbf{c}_i$. The utterance is accepted as one from speaker i if the similarity is exceeds a given threshold. We repeat the experiments with the same enrolment set and replace the verification set with samples generated by each adaptation method under different data size settings.

In our setup we fix the enrolment set together with the speaker verification model from [169], and study the performance of different verification sets that are either from real utterances or generated by a TTS system. Table 4.3 lists the equal error



(a) LibriSpeech, 1 min **(b)** VCTK, 1 min

(c) LibriSpeech, 10 sec **(d)** VCTK, 10 sec

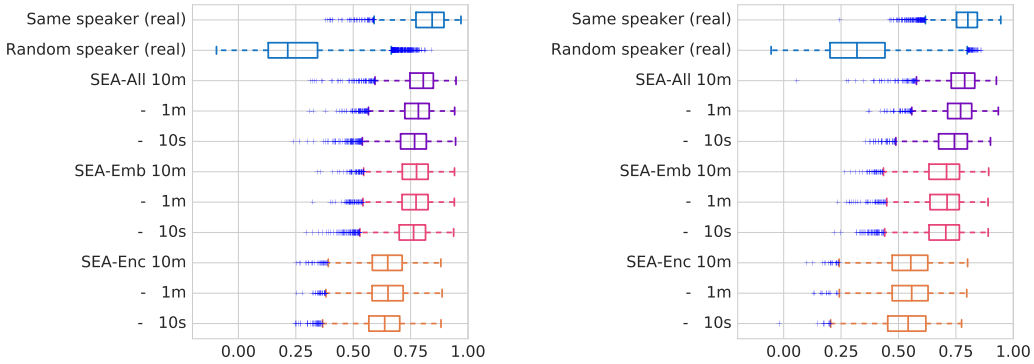
Figure 4.6: Detection error trade-off (DET) curve for speaker verification, using the TI-SV speaker verification model [169]. The utterances were generated using 1 minute or 10 seconds of utterance from LibriSpeech and VCTK. EER is marked with a dot.

Dataset	LibriSpeech			VCTK		
Real utterance	2.47			2.79		
Adaptation data size	*10s*	*1m*	*<5m*	*10s*	*1m*	*<10m*
SEA-All (ours)	**3.17**	**2.47**	**1.85**	**7.34**	**5.02**	**4.33**
SEA-Emb (ours)	3.26	2.92	2.74	10.18	9.91	10.24
SEA-Enc (ours)	10.73	9.77	9.42	27.20	25.34	25.23

Table 4.3: Equal error rate (EER) of real and few-shot adapted voice samples for evaluation of voice similarity. Varying adaptation dataset sizes were considered.

rate (EER) of the verification model with real and generated verification utterances, and Figure 4.6 shows the detection error trade-off (DET) curves for a more thorough inspection. Figure 4.6 shows the adaptation models with different data size settings.

SEA-All outperforms the other two approaches, and the error rate decreases clearly with the size of demonstration data, while SEA-Emb model performs better



(a) LibriSpeech **(b)** VCTK

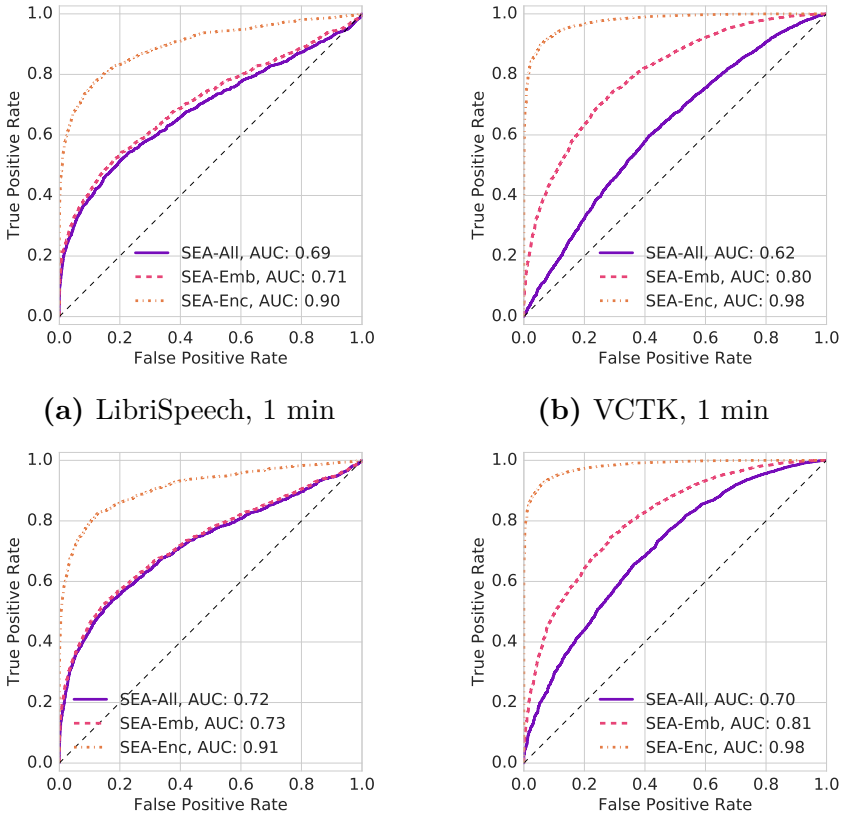
Figure 4.7: Cosine similarity of real and generated utterances to the real enrollment set.

than SEA-Enc. Additionally, the benefit of more demonstration data is less significant than for SEA-All in both of these models. Noticeably, the EER of SEA-All is even lower than the real utterance on the LibriSpeech dataset with sufficient adaptation data. A possible explanation is that the generated samples might be concentrated closer to the centroid of a speaker's embeddings than real speech with larger variance across utterances. Further, instead of focusing on the single EER point estimates (dots in Figure 4.6), it is more informative to consider the similarity between the whole DET curves corresponding to the different generative models and the DET curve of the real data in Figure 4.6.

Discerning real from generated utterances

In this section, we compare the generated samples and the real utterances of the speaker being imitated. Figure 4.7 shows the box-plot of the cosine similarity between the embedding centroids of test speakers' enrollment set and (1) real utterances from the same speaker, (2) real utterances from a different speaker, and (3) generated utterances adapted to the same speaker. Consistent with the observations from the previous subsection, SEA-All performs best.

We further consider an adversarial scenario for speaker verification. In contrast to the previous standard speaker verification setup where we now select a verification utterance with either a real utterance from the *same* speaker or a synthetic sample



(a) LibriSpeech, 1 min **(b)** VCTK, 1 min

(c) LibriSpeech, 10 sec **(d)** VCTK, 10 sec

Figure 4.8: ROC curve for real versus generated utterance detection. The utterances were generated using 1 minute or 10 seconds of utterance from LibriSpeech and VCTK. Lower curve indicates that the verification system is having a harder time distinguishing real from generated samples.

from a model adapted to the *same* speaker. Under this setup, the speaker verification system is challenged by synthetic samples and acts as a classifier for real versus generated utterances. The ROC curve of this setup is shown in Figure 4.8 and the models are using variable data size settings. If the generated samples are indistinguishable from real utterances, the ROC curve approaches the diagonal line (that is, the verification system fails to separate real and generated voices). Importantly, SEA-All manages to confuse the verification system especially for the VCTK dataset where the ROC curve is almost inline with the diagonal line with an AUC of 0.56.

4.6 Conclusion

This chapter presented three variants of meta-learning for sample efficient adaptive TTS. The adaptation method that fine-tunes the entire model, with the speaker embedding vector first optimised, shows impressive performance even with only 10 seconds of audio from new speakers. When adapted with a few minutes of data, our model matches the state-of-the-art performance in sample naturalness. Moreover, it outperforms other recent works in matching the new speaker's voice. We also demonstrated that the generated samples achieved a similar level of voice similarity to real utterances from the same speaker, when measured by a text independent speaker verification model. Our work considers the adaptation to new voices with clean, high-quality training data collected in a controlled environment. The few-shot learning of voices with noisy data is beyond the scope of this work and remains a challenging open research problem, which limits potential misuse. We hope that this technology will be used for good purposes, to improve the human-computer interaction, and expand to important medical applications.

Finally, in this chapter we showed the effectiveness of meta-learning in speech adaptation and the potential it has to medical applications. Our research has focused on the emergence, the recognition, and the synthesis of communication. Within these selected areas of the process of communication, our work introduced state-of-the-art solutions using deep learning. The last chapter outlines future research directions and discusses the potential impact of such technologies to our society.

5 Conclusions and future directions

In this thesis, we introduced state-of-the-art solutions within selected areas of the process of communication using deep learning. In this final chapter, we outline the conclusions of our work and highlight future research directions. Finally, we emphasise the importance of ethical and moral issues which we believe society and its legal systems should adapt to, and include in their policies.

5.1 Conclusions

By using Taylor's model of communication as a guiding framework, we were able to isolate three discrete subprocesses of communication, emergence, recognition, and synthesis. This thesis made state-of-the-art contributions in each of the subprocesses using deep learning.

First, we considered the problem of multiple agents sensing and acting in environments to maximise their shared utility. In these environments, agents must learn communication protocols to share information that is needed to solve the tasks. With the methods presented we have shown that such protocols can be learnt using reinforcement learning, and that the learning process can be improved by introducing differentiable communication and end-to-end training.

Second, we showed the effectiveness of end-to-end learning for the recognition of visual verbal communication. More specifically, we presented two novel models of verbal communication recognition by tackling sentence-level lipreading for the first time end-to-end. Our models outperform human lipreaders, aptly illustrating the potential of such technologies for improving speech recognition in noisy environments, and aiding communication for people affected by aphonia, dysphonia, and other medical conditions.

Third, we investigated whether meta-learning can help generating verbal communication and text-to-speech synthesis using a limited amount of data. We introduced a novel approach that achieves state-of-the-art results in both sample naturalness and voice similarity based on merely a few minutes of audio data from newly introduced speakers. Our results indicate a strong potential for medical applications using voice synthesis.

In the following paragraphs, we discuss additional areas and future directions that our work could be applied to, in order to maximise its potential for making a lasting impact on communications processes.

5.2 Future directions

Emergence

Differentiable communication allows optimal communication protocols to emerge. These protocols could be distilled to human interpretable rulesets for instance using decision trees or online learning, in order to render them dynamic and allow them to adjust to structural or hardware changes. These methods could have a great impact on routing and load balancing protocols for computer or mobile networks such as 5G. Current advances in Software-defined Networking (SDN) allow a programmatically dynamic network configuration to improve the network's performance. Within the network, the nodes operate in a partially observable environment, where they can observe local traffic and share statistics about it with other nodes. Load balancing protocols rely on those statistics. Differentiable communication protocols could be used to replace these shared statistics and efficiently take action for managing packet routing. An additional field of impact is identified in the systems of swarm intelligence, where agents act in accordance to a decentralised, self-organised behaviour, as is for instance the case with self-driving cars or drones. In both cases, the agents operating in the real world must deal with partial observability with a limited bandwidth communication channel. Thus, by using differentiable communication, these agents could create protocols to share their observations efficiently and improve the quality of service.

Recognition

We should aim to advance medical applications of visual speech recognition, to improve the quality of the stay of hospitalised patients affected by conditions causing aphonia or dysphonia. Such patients naturally have the need to communicate with doctors and the desire to communicate with their loved ones. Currently, their communication relies on visual cues and hiring professional lipreaders. Automatic lipreading systems could have a notable impact in such circumstances. Having solved and perfected end-to-end visual speech recognition, the challenge now is the efficient combination with an audio signal in order to achieve full audio-visual

speech recognition. This could be achieved using an early or late fusion of audio and visual features, and could therefore also be of use for speech recognition in noisy environments such as public places and cars, in addition to improving communication applications, subtitling and biometrics.

Synthesis

Medical research is a critical area for future work in this field too. More specifically, speech synthesis models relying on limited amounts of data would be of great use to patients affected by voice-disorders requiring assistive text-to-speech technologies to speak. For instance, patients with Amyotrophic Lateral Sclerosis (ALS) could strongly benefit from these communication technologies. Further research should focus on relaxing the need for the generation of linguistic features (durations, prosody etc.), while maintaining the quality of the original WaveNet architecture. Another potential direction is indicated as the study of the performance of meta-learning algorithms on noisy and low-quality data. Further applications could include personalised human-computer-interfaces such as personal assistants, and the generation of entertainment shows with voices that have never been heard before.

5.3 Epilogue

In conclusion, we wish to stress the point that novel technologies are, ultimately, tools. The more advanced the technology, the more sophisticated the tool. Moreover, tools are intended to be used, and such uses can be directed towards the improvement of quality of life. Regrettably, the same tools can also be misused instead: the typical example being that a hammer, which can be used to build a house, or to destroy one [208]. Part of the legal system's role in society is to act as a filter which forbids and penalises misuse. By emphasising the instrumental and assistive nature of the work proposed in the present thesis, we are also highlighting

the ethical and moral issues which current societies and their legal systems must consider in the drafting of policy.

Our success in understanding intelligence is tightly coupled with our ability to analyse and model human intelligence, and thus relies on utilising real-world human data. This is necessary if we are to create future artificial general intelligence systems that are compassionate, capable of counterfactual reasoning, collaborative and kind. Currently, by using real-world data, machine learning models can surpass human performance in a number of specific tasks, and vastly increase the potential for improving the quality of our lives. To protect this potential, it is important to focus on the privacy, security and legal aspects of novel technologies as the ones presented above. Taking as an example our work on visual speech recognition, and despite the references of lipreading in science fiction novels and movies, any expert can understand that there are no privacy threats imposed, but there are legal requisites introduced. For instance, owing to the difficulty of obtaining a continually front-on view of a face at a sufficiently high resolution without an individual's consent, such models are not suited in scenarios of mass surveillance. Although targeted surveillance is possible, there is a plethora of more effective ways, which are also robust to a hand covering of the mouth. This leads to the conclusion that, regardless of the means, mass surveillance by any party without the consent of the people being viewed should be illegal. In the case of synthesis, detecting, signing and verifying so-called "fake media" in today's media outlets, social platforms and popular culture is essential. However, this need is orthogonal to research on generative models of human data, as often the issues stem from editing "real media" with the simplest manipulation methods, e.g. removing frames, accelerating the play speed at selected parts, and others [209]. Thus, it is vital to raise public awareness of such matters, prohibit misuse of unverified data and motivate research on digital signing of data, as we do with emails.

In these conclusive sections, we have therefore highlighted the beneficial technologies stemming from our research to society. It is a mindful and conscious statement of this work's commitment to enhancing the communal aspect of communication,

something embedded in its very etymology, which lays in the potential of future extensions of this research. The research discussions that took place throughout the writing of this thesis and emerged through the interactions with external parties consistently indicated the obligation to sharing these premises and caveats with the general public, the importance of open information sharing, and the need for current legal systems to rapidly adjust to new technologies. It is our hope that this work will act as a catalyst for such discussions, motivate future interdisciplinary work, and trace the initial steps towards the development of novel technologies for the greater good of society.

Bibliography

[1] J. N. Foerster*, Y. Assael*, N. de Freitas, and S. Whiteson. Learning to communicate to solve riddles with deep distributed recurrent Q-networks. In *International Joint Conference on Artificial Intelligence Workshop*, 2016.

[2] J. N. Foerster*, Y. Assael*, N. de Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.

[3] Y. Assael*, B. Shillingford*, S. Whiteson, and N. de Freitas. LipNet: End-to-end sentence-level lipreading. In *GPU Technology Conference*, 2017.

[4] B. Shillingford*, Y. Assael*, M. W. Hoffman, T. Paine, C. Hughes, U. Prabhu, H. Liao, H. Sak, K. Rao, L. Bennett, M. Mulville, M. Denil, B. Coppin, B. Laurie, A. Senior, and N. de Freitas. Large-scale visual speech recognition. In *INTERSPEECH*, 2019.

[5] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie, C. Gulcehre, A. van den Oord, O. Vinyals, and N. de Freitas. Sample efficient adaptive text-to-speech. In *International Conference on Learning Representations*, 2019.

[6] M. J. Assael, K. D. Antoniadis, I. N. Metaxa, S. K. Mylona, Y. Assael, J. Wu, and M. Hu. A novel portable absolute transient hot-wire instrument for the measurement of the thermal conductivity of solids. *International Journal of Thermophysics*, 36(10-11):3083–3105, 2015.

[7] H. Mossalam, Y. Assael, D. M. Roijers, and S. Whiteson. Multi-objective deep reinforcement learning. In *Advances in Neural Information Processing Systems Deep Reinforcement Learning Workshop*, 2016.

[8] R. Ponte Costa*, Y. Assael*, B. Shillingford*, N. de Freitas, and T. Vogels. Cortical microcircuits as gated-recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 272–283, 2017.

[9] Y. Assael, K. D. Antoniadis, and M. J. Assael. From analog timers to the era of machine learning: The case of the transient hot-wire technique. In *Thermophysics*, volume 1866, page 020001. AIP Publishing, 2017.

[10] N. D. Goumagias, D. Hristu-Varsakelis, and Y. Assael. Using deep Q-learning to understand the tax evasion behavior of risk-averse firms. *Expert Systems with Applications*, 101:258–270, 2018.

[11] A. Gupta, B. Shillingford, Y. Assael, and T. C. Walters. Speech bandwidth extension with wavenet. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2019.

[12] B. Shillingford, Y. Assael, and M. Denil. Interactive decoding of words from visual speech recognition models. In *arXiv preprint*, 2019.

[13] T. Makino, H. Liao, Y. Assael, B. Shillingford, B. Garcia, O. Braga, and O. Siohan. Recurrent neural network transducer for audio-visual speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE, 2019.

[14] Y. Assael*, T. Sommerschield*, and J. Prag. Restoring ancient text using deep learning: a case study on Greek epigraphy. In *Empirical Methods in Natural Language Processing*, 2019.

[15] S. Legg, M. Hutter, et al. A collection of definitions of intelligence. *Frontiers in Artificial Intelligence and applications*, 157:17, 2007.

[16] P. Taylor. *Text-to-Speech Synthesis*. Cambridge University Press, 1st edition, 2009.

[17] Y. N. Harari. *Sapiens: A brief history of humankind*. Random House, 2014.

[18] S. W. Littlejohn and K. A. Foss. *Theories of human communication*. Waveland press, 2010.

[19] M. Tomasello. *Origins of human communication*. MIT press, 2010.

[20] D. Beard. Forum: On the history of communication studies. *Quarterly Journal of Speech*, 93(3):344, 2007.

[21] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[22] R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Zidek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, D. T. Jones, D. Silver, K. Kavukcuoglu, D. Hassabis, and A. W. Senior. De novo structure prediction with deeplearning based scoring. *Annual Review of Biochemistry*, 77:363–382, 2018.

[23] S. Reed and N. de Freitas. Neural programmer-interpreters. In *International Conference on Learning Representations*, 2016.

[24] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471, 2016.

[25] M. Bošnjak, T. Rocktäschel, J. Naradowsky, and S. Riedel. Programming with a differentiable forth interpreter. In *International Conference on Machine Learning*, pages 547–556, 2017.

[26] M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. In *International Conference on Learning Representations*, 2018.

[27] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.

[28] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[29] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *SSW*, 125, 2016.

[30] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision*. IEEE, 2017.

[31] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

[32] O. Vinyals, I. Babuschkin, J. Chung, M. Mathieu, M. Jaderberg, W. M. Czarnecki, A. Dudzik, A. Huang, P. Georgiev, R. Powell, T. Ewalds, D. Horgan, M. Kroiss, I. Danihelka, J. Agapiou, J. Oh, V. Dalibard, D. Choi, L. Sifre, Y. Sulsky, S. Vezhnevets, J. Molloy, T. Cai, D. Budden, T. Paine, C. Gulcehre, Z. Wang, T. Pfaff, T. Pohlen, Y. Wu, D. Yogatama, J. Cohen, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, C. Apps, K. Kavukcuoglu, D. Hassabis, and D. Silver. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. `https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/`, 2019. Accessed on 2019-04-24.

[33] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[34] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[35] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[36] P. Cobley and P. J. Schulz. *Theories and models of communication*, volume 1. Walter de Gruyter, 2013.

[37] C. K. Ogden and I. A. Richards. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*, volume 29. K. Paul, Trench, Trubner & Company, Limited, 1923.

[38] J. Fiske. *Introduction to communication studies*. Routledge, 2010.

[39] R. T. Craig. Communication theory as a field. *Communication theory*, 9(2): 119–161, 1999.

[40] Aristotle. Rhetoric, 367–322 BC.

[41] R. Whately. *Elements of Rhetoric: Comprising an Analysis of the Laws of Moral Evidence and of Persuasion: with Rules for Argumentative Composition and Elocution*. Longmans, Green, Reader, and Dyer, 1873.

[42] U. Narula. *Communication models*. Atlantic Publishers & Dist, 2006.

[43] H. D. Lasswell. The structure and function of communication in society. *The communication of ideas*, 37:215–228, 1948.

[44] C. E. Shannon and W. Weaver. The mathematical theory of information. Technical report, University of Illinois Press, 1949.

[45] W. Schramm. How communication works. *The process and effects of mass communication*, pages 3–26, 1954.

[46] M. L. DeFleur. *Theories of mass communication*. Pearson, 1970.

[47] E. Griffin. *A first look at communication theory*. McGraw-Hill, 2006.

[48] B. H. Westley and M. S. MacLean Jr. A conceptual model for communications research. *Journalism Quarterly*, 34(1):31–38, 1957.

[49] S. Számadó and E. Szathmáry. Language evolution. *PLoS biology*, 2(10):346, 2004.

[50] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

[51] L. Kraemer and B. Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.

[52] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[53] M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *International Conference on Machine Learning*, 1993.

[54] F. S. Melo, M. T. Spaan, and S. J. Witwicki. Querypomdp: Pomdp-based communication in multiagent systems. In *European Workshop on Multi-Agent Systems*, pages 189–204. Springer, 2011.

[55] L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

[56] T. Kasai, H. Tenmoto, and A. Kamiya. Learning of communication codes in multi-agent reinforcement learning problem. In *Soft Computing in Industrial Applications*, pages 1–6. IEEE, 2008.

[57] C. L. Giles and K. C. Jim. Learning communication for multi-agent systems. In *Innovative Concepts for Agent-Based Systems*, pages 377–390. Springer, 2002.

[58] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: a recurrent neural network for image generation. In *International Conference on International Conference on Machine Learning*, pages 1462–1471, 2015.

[59] S. Sukhbaatar, R. Fergus, et al. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, pages 2244–2252, 2016.

[60] M. Werning, W. Hinzen, and E. Machery. *The Oxford handbook of compositionality*. Oxford Handbooks in Linguistic, 2012.

[61] M. Courbariaux and Y. Bengio. BinaryNet: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv preprint arXiv:1602.02830*, 2016.

[62] G. Hinton and R. Salakhutdinov. Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science*, 3(1):74–91, 2011.

[63] R. S. Sutton and A. G. Barto. *Introduction to reinforcement learning*. MIT Press, 1998.

[64] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente. Multiagent cooperation and competition with deep reinforcement learning. *PloS one*, 12(4):1–15, 2017.

[65] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, New York, 2009.

[66] E. Zawadzki, A. Lipson, and K. Leyton-Brown. Empirically evaluating multiagent learning algorithms. *arXiv preprint 1401.8074*, 2014.

[67] M. Hausknecht and P. Stone. Deep recurrent Q-learning for partially observable MDPs. In *Association for the Advancement of Artificial Intelligence Fall Symposia*, pages 29–37, 2015.

[68] K. Narasimhan, T. D. Kulkarni, and R. Barzilay. Language understanding for textbased games using deep reinforcement learning. In *Empirical Methods in Natural Language Processing*, 2015.

[69] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[70] C. Sammut and G. I. Webb. *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.

[71] A. Naik, R. Shariff, N. Yasui, and R. S. Sutton. Discounted reinforcement learning is not an optimization problem. *arXiv preprint arXiv:1910.02140*, 2019.

[72] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012.

[73] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014.

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

[75] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[76] W. Wu. 100 prisoners and a lightbulb. Technical report, OCF, UC Berkeley, 2002.

[77] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[78] M. Studdert-Kennedy. How did language go discrete? In M. Tallerman, editor, *Language Origins: Perspectives on Evolution*, chapter 3. Oxford University Press, 2005.

[79] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip reading sentences in the wild. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[80] A. Thanda and S. M. Venkatesan. Audio visual speech recognition using deep recurrent neural networks. In F. Schwenker and S. Scherer, editors, *Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction*, pages 98–109. Springer, 2017.

[81] A. Koumparoulis, G. Potamianos, Y. Mroueh, and S. J. Rennie. Exploring ROI size in deep learning based lipreading. In *International Conference on Auditory-Visual Speech Processing*, 2017.

[82] J. S. Chung and A. Zisserman. Lip reading in profile. In *British Machine Vision Conference*, 2017.

[83] K. Xu, D. Li, N. Cassimatis, and X. Wang. Lcanet: End-to-end lipreading with cascaded attention-ctc. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 548–555. IEEE, 2018.

[84] J. S. Chung and A. Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, 2016.

[85] M. Wand, J. Koutnik, and J. Schmidhuber. Lipreading with long short-term memory. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 6115–6119. IEEE, 2016.

[86] T. Stafylakis and G. Tzimiropoulos. Combining residual networks with LSTMs for lipreading. In *INTERSPEECH*, pages 3652–3656, 2017.

[87] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *International Conference on Machine Learning*, pages 689–696, 2011.

[88] C. Sui, M. Bennamoun, and R. Togneri. Listening with your eyes: Towards a practical visual speech recognition system using deep Boltzmann machines. In *IEEE International Conference on Computer Vision*, pages 154–162. IEEE, 2015.

[89] H. Ninomiya, N. Kitaoka, S. Tamura, Y. Iribe, and K. Takeda. Integration of deep bottleneck features for audio-visual speech recognition. In *International Speech Communication Association*, 2015.

[90] S. Petridis and M. Pantic. Deep complementary bottleneck features for visual speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2304–2308. IEEE, 2016.

[91] S. Petridis, Y. Wang, Z. Li, and M. Pantic. End-to-end multi-view lipreading. In *British Machine Vision Conference*, 2017.

[92] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata. Lipreading using convolutional neural network. In *INTERSPEECH*, pages 1149–1153, 2014.

[93] O. Koller, H. Ney, and R. Bowden. Deep learning of mouth shapes for sign language. In *International Machine Vision Conference Workshop on Assistive Computer Vision and Robotics*, pages 85–91, 2015.

[94] I. Almajai, S. Cox, R. Harvey, and Y. Lan. Improved speaker independent lip reading using speaker adaptive training and deep neural networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2722–2726. IEEE, 2016.

[95] Y. Takashima, R. Aihara, T. Takiguchi, Y. Ariki, N. Mitani, K. Omori, and K. Nakazono. Audio-visual speech recognition using bimodal-trained bottleneck features for a person with severe hearing loss. In *INTERSPEECH*, pages 277–281, 2016.

[96] M. Wand and J. Schmidhuber. Improving speaker-independent lipreading with domain-adversarial training. In *INTERSPEECH*, 2017.

[97] HCUPnet. Hospital inpatient national statistics. `https://hcupnet.ahrq.gov/`, 2014. Accessed: 2018-04-23.

[98] The Health Foundation. New safety collaborative will improve outcomes for patients with tracheostomies. `http://www.health.org.uk/news/new-safety-collaborative-will-improve-outcomes-patients-tracheostomies`, 2014. Accessed: 2018-04-23.

[99] M. A. Morris and A. N. Kho. Silence in the EHR: Infrequent documentation of aphonia in the electronic health record. *BMC Health Services Research*, 14 (1):425, 2014.

[100] S. M. Chu and T. S. Huang. Bimodal speech recognition using coupled hidden markov models. In *International Conference on Spoken Language Processing*, 2000.

[101] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, 2002.

[102] V. Pitsikalis, A. Katsamanis, G. Papandreou, and P. Maragos. Adaptive multimodal fusion by uncertainty compensation. In *International Conference on Spoken Language Processing*, 2006.

[103] P. Lucey and S. Sridharan. Patch-based representation of visual speech. In *HCSNet Workshop on Use of Vision in Human-Computer Interaction*, pages 79–85, 2006.

[104] G. Papandreou, A. Katsamanis, V. Pitsikalis, and P. Maragos. Multimodal fusion and learning with uncertain features applied to audiovisual speech recognition. In *IEEE Workshop on Multimedia Signal Processing*, pages 264–267. IEEE, 2007.

[105] G. Zhao, M. Barnard, and M. Pietikainen. Lipreading with local spatiotemporal descriptors. *IEEE Transactions on Multimedia*, 11(7):1254–1265, 2009.

[106] M. Gurban and J.-P. Thiran. Information theoretic feature extraction for audio-visual speech recognition. *IEEE Transactions on Signal Processing*, 57 (12):4765–4776, 2009.

[107] G. Papandreou, A. Katsamanis, V. Pitsikalis, and P. Maragos. Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(3):423–435, 2009.

[108] A. J. Goldschen, O. N. Garcia, and E. D. Petajan. Continuous automatic speech recognition by lipreading. In *Motion-Based recognition*, pages 321–343. Springer, 1997.

[109] G. Potamianos, E. Cosatto, H. P. Graf, and D. B. Roe. Speaker independent audio-visual database for bimodal asr. In *Audio-Visual Speech Processing: Computational & Cognitive Science Approaches*, 1997.

[110] G. Potamianos and H. P. Graf. Discriminative training of hmm stream exponents for audio-visual speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3733–3736. IEEE, 1998.

[111] G. Potamianos, H. P. Graf, and E. Cosatto. An image transform approach for hmm based automatic lipreading. In *IEEE Conference on Image Processing*, pages 173–177. IEEE, 1998.

[112] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, and A. Mashari. Audio visual speech recognition. Technical report, IDIAP, 2000.

[113] S. Gergen, S. Zeiler, A. H. Abdelaziz, R. Nickel, and D. Kolossa. Dynamic stream weighting for turbo-decoding-based audiovisual ASR. In *INTERSPEECH*, pages 2135–2139, 2016.

[114] K. Paleček. Utilizing lipreading in large vocabulary continuous speech recognition. In *Speech and Computer*, pages 767–776. Springer, 2017.

[115] A. B. Hassanat. Visual speech recognition. In *Speech and Language Technologies*. InTech, 2011.

[116] H. L. Bear and R. Harvey. Decoding visemes: Improving machine lip-reading. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2009–2013. IEEE, 2016.

[117] G. Potamianos, C. Neti, J. Luettin, and I. Matthews. Audio-visual automatic speech recognition: An overview. *Issues in Visual and Audio-Visual Speech Processing*, 22:23, 2004.

[118] Z. Zhou, G. Zhao, X. Hong, and M. Pietikäinen. A review of recent advances in visual speech decoding. *Image and Vision Computing*, 32(9):590–605, 2014.

[119] M. Cooke, J. Barker, S. Cunningham, and X. Shao. An audio-visual corpus for speech perception and automatic speech recognition. *Journal of the Acoustical Society of America*, 120(5):2421–2424, 2006.

[120] J. S. Chung and A. Zisserman. Out of time: Automated lip sync in the wild. In *Asian Conference on Computer Vision Workshop on Multi-view Lip-reading*, 2016.

[121] T. Le Cornu and B. Milner. Generating intelligible audio speech from visual speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 25 (9):1751–1761, 2017.

[122] A. Ephrat and S. Peleg. Vid2speech: Speech reconstruction from silent video. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 5095–5099. IEEE, 2017.

[123] H. Akbari, H. Arora, L. Cao, and N. Mesgarani. Lip2audspec: Speech reconstruction from silent lip movements video. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2516–2520. IEEE, 2018.

[124] A. Gabbay, A. Shamir, and S. Peleg. Visual speech enhancement using noise-invariant training. *arXiv preprint arXiv:1711.08789*, 2017.

[125] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics*, 37(4):112, 2018.

[126] T. Afouras, J. S. Chung, and A. Zisserman. The conversation: Deep audio-visual speech enhancement. In *INTERSPEECH*, pages 3244–3248, 2018.

[127] C. Chambers, A. Raniwala, F. Perry, S. Adams, R. R. Henry, R. Bradshaw, and N. Weizenbaum. Flumejava: easy, efficient data-parallel pipelines. In *Sigplan Notices*, volume 45, pages 363–375. ACM, 2010.

[128] H. Liao, E. McDermott, and A. Senior. Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 368–373. IEEE, 2013.

[129] H. Liao, G. Pundak, O. Siohan, M. K. Carroll, N. Coccaro, Q.-M. Jiang, T. N. Sainath, A. Senior, F. Beaufays, and M. Bacchiani. Large vocabulary automatic speech recognition for children. In *International Speech Communication Association*, 2015.

[130] V. Kuznetsov, H. Liao, M. Mohri, M. Riley, and B. Roark. Learning n-gram language models from uncertain data. In *INTERSPEECH*, pages 2323–2327, 2016.

[131] H. Soltau, H. Liao, and H. Sak. Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition. In *INTERSPEECH*, pages 3707–3711, 2017.

[132] A. Salcianu, A. Golding, A. Bakalov, C. Alberti, D. Andor, D. Weiss, E. Pitler, G. Coppola, J. Riesa, K. Ganchev, M. Ringgaard, N. Hua, R. McDonald, S. Petrov, S. Istrate, and T. Koo. Compact language detector v3. `https://github.com/google/cld3`, 2018.

[133] J. C. Wells. Computer-coding the IPA: A proposed extension of SAMPA. Technical report, University College London, 1995.

[134] J. Mas and G. Fernandez. Video shot boundary detection based on color histogram. *Notebook Papers TRECVID, NIST*, 15, 2003.

[135] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Conference on Computer Vision and Pattern Recognition*, pages 815–823. IEEE, 2015.

[136] A. Torfi, S. M. Iranmanesh, N. Nasrabadi, and J. Dawson. 3d convolutional neural networks for cross audio-visual matching recognition. *IEEE Access*, 5: 22081–22091, 2017.

[137] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[138] K. Mase and A. Pentland. Automatic lipreading by optical-flow analysis. *Systems and Computers in Japan*, 22(6):67–76, 1991.

[139] M. S. Gray, J. R. Movellan, and T. J. Sejnowski. Dynamic features for visual speechreading: A systematic comparison. In *Advances in Neural Information Processing Systems*, pages 751–757, 1997.

[140] T. Yoshinaga, S. Tamura, K. Iwano, and S. Furui. Audio-visual speech recognition using lip movement extracted from side-face images. In *International Conference on Audio-Visual Speech Processing*, 2003.

[141] S. Tamura, K. Iwano, and S. Furui. Multi-modal speech recognition using optical-flow analysis for lip images. In *Real World Speech Processing*, pages 43–50. Springer, 2004.

[142] S.-L. Wang, A. W.-C. Liew, W. H. Lau, and S. H. Leung. An automatic lipreading system for spoken digits with limited training data. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(12):1760–1765, 2008.

[143] A. A. Shaikh, D. K. Kumar, W. C. Yau, M. C. Azemin, and J. Gubbi. Lip reading using optical flow and support vector machines. In *IEEE International Congress on Image and Signal Processing*, volume 1, pages 327–330. IEEE, 2010.

[144] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[145] Y. Wu and K. He. Group normalization. In *European Conference on Computer Vision*, pages 3–19, 2018.

[146] A. Graves, S. Fernandez, F. J. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural nets. In *International Conference on Machine Learning*, pages 369–376, 2006.

[147] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182, 2016.

[148] K. Rao, H. Sak, and R. Prabhavalkar. Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pages 193–199. IEEE, 2017.

[149] R. Prabhavalkar, K. Rao, T. Sainath, B. Li, L. Johnson, and N. Jaitly. A comparison of sequence-to-sequence models for speech recognition. In *INTERSPEECH*, 2017.

[150] M. Mohri, F. Pereira, and M. Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

[151] Y. Miao, M. Gowayyed, and F. Metze. Eesen: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 167–174. IEEE, 2015.

[152] I. McGraw, R. Prabhavalkar, R. Alvarez, M. G. Arenas, K. Rao, D. Rybach, O. Alsharif, H. Sak, A. Gruenstein, F. Beaufays, et al. Personalized speech recognition on mobile devices. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 5955–5959. IEEE, 2016.

[153] A. Salomaa and M. Soittola. Automata—theoretic aspects of formal power series. *Bull. Amer. Math. Soc*, 1:675–678, 1979.

[154] J. Berstel Jr and C. Reutenauer. *Rational series and their languages*. Springer-Verlag, 1988.

[155] W. Kuich and A. Salomaa. Semirings, automata, languages. 1985.

[156] J. L. Pech-Pacheco, G. Cristóbal, J. Chamorro-Martinez, and J. Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: A comparative study. In *IEEE Pattern Recognition*, volume 3, pages 314–317. IEEE, 2000.

[157] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations Workshop*, 2014.

[158] T. Afouras, J. S. Chung, and A. Zisserman. LRS3-TED: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496*, 2018.

[159] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Deep audio-visual speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[160] E. Cooper, E. Li, and J. Hirschberg. Characteristics of text-to-speech and other corpora. In *International Conference on Speech Prosody*, pages 690–694, 2018.

[161] L. C. Bardach and D. S. Newman. Augmentative and alternative communication in als. *Perspectives on Augmentative and Alternative Communication*, 12 (5):14–21, 2003.

[162] A. Iida and N. Campbell. Speech database design for a concatenative text-to-speech synthesis system for individuals with communication disorders. *International Journal of Speech Technology*, 6(4):379–392, 2003.

[163] S. Kawahara, M. Homma, T. Yoshimura, and T. Arai. Myvoice: Rescuing voices of als patients. *Acoustical Science and Technology*, 37(5):202–210, 2016.

[164] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[165] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[166] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. In *Computer Vision and Pattern Recognition*, 2015.

[167] T. Dutoit. *An Introduction to Text-to-speech Synthesis*. Kluwer Academic Publishers, Norwell, MA, USA, 1997. ISBN 0-7923-4498-7.

[168] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis. Parallel WaveNet: Fast high-fidelity speech synthesis. In *International Conference on Machine Learning*, volume 80 of *Machine Learning Research*, pages 3918–3926. PMLR, 2018.

[169] L. Wan, Q. Wang, A. Papir, and I. L. Moreno. Generalized end-to-end loss for speaker verification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 4879–4883. IEEE, 2018.

[170] H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak. Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices. In *INTERSPEECH*, pages 2273–2277, 2016.

[171] P. Ebden and R. Sproat. The kestrel tts text normalization system. *Natural Language Engineering*, 21(3):333–353, 2015.

[172] A. De Cheveigné and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111 (4):1917–1930, 2002.

[173] R. Dall, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda. Redefining the linguistic context feature set for hmm and dnn tts through position and parsing. In *INTERSPEECH*, pages 2851–2855, 2016.

[174] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang. Deep speaker feature learning for text-independent speaker verification. In *INTERSPEECH*, pages 1542–1546, 2017.

[175] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, pages 1842–1850, 2016.

[176] P. Shyam, S. Gupta, and A. Dukkipati. Attentive recurrent comparators. In *International Conference on Machine Learning*, pages 3173–3181, 2017.

[177] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[178] O. Abdel-Hamid and H. Jiang. Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7942–7946. IEEE, 2013.

[179] T. Pohlen, B. Piot, T. Hester, M. G. Azar, D. Horgan, D. Budden, G. Barth-Maron, H. van Hasselt, J. Quan, M. Večerík, et al. Observe and look further: Achieving consistent performance on atari. *arXiv preprint arXiv:1805.11593*, 2018.

[180] Y. Aytar, T. Pfaff, D. Budden, T. Paine, Z. Wang, and N. de Freitas. Playing hard exploration games by watching youtube. In *Advances in Neural Information Processing Systems*, pages 2930–2941, 2018.

[181] H. F. Harlow. The formation of learning sets. *Psychological review*, 56(1):51, 1949.

[182] S. Thrun and L. Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[183] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.

[184] Y. Chen, M. W. Hoffman, S. G. Colmenarejo, M. Denil, T. P. Lillicrap, M. Botvinick, and N. Freitas. Learning to learn without gradient descent by gradient descent. In *International Conference on Machine Learning*, pages 748–756, 2017.

[185] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representations*, 2016.

[186] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

[187] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. One-shot visual imitation learning via meta-learning. In *Conference on Robot Learning*, pages 357–368, 2017.

[188] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. In *International Conference on Learning Representations Workshop*, 2018.

[189] S. Bartunov and D. P. Vetrov. Fast adaptation in generative models with generative matching networks. In *International Conference on Learning Representations Workshop*, 2017.

[190] J. Bornschein, A. Mnih, D. Zoran, and D. J. Rezende. Variational memory addressing in generative models. In *Advances in Neural Information Processing Systems*, pages 3923–3932, 2017.

[191] D. J. Rezende, S. Mohamed, I. Danihelka, K. Gregor, and D. Wierstra. One-shot generalization in deep generative models. In *International Conference on Machine Learning*, pages 1521–1529, 2016.

[192] S. Reed, Y. Chen, T. Paine, A. van den Oord, S. M. Eslami, D. Rezende, O. Vinyals, and N. de Freitas. Few-shot autoregressive density estimation: Towards learning to learn distributions. In *International Conference on Learning Representations*, 2018.

[193] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756, 2016.

[194] J. Veness, T. Lattimore, A. Bhoopchand, A. Grabska-Barwinska, C. Mattern, and P. Toth. Online learning with gated linear networks. *arXiv preprint arXiv:1712.01897*, 2017.

[195] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. *arXiv preprint arXiv:1803.09047*, 2018.

[196] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous. Tacotron: Towards end-to-end speech synthesis. In *INTERSPEECH*, pages 4006–4010, 2017.

[197] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou. Deep voice 2: Multi-speaker neural text-to-speech. In *Advances in Neural Information Processing Systems*, pages 2962–2970, 2017.

[198] S. Ö. Arık, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, et al. Deep voice: Real-time neural text-to-speech. In *International Conference on Machine Learning*, pages 195–204, 2017.

[199] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller. Deep voice 3: 2000-speaker neural text-to-speech. In *International Conference on Learning Representations*, 2018.

[200] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio. Char2wav: End-to-end speech synthesis. In *International Conference on Learning Representations Workshop*, 2017.

[201] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani. Voiceloop: Voice fitting and synthesis via a phonological loop. In *International Conference on Learning Representations*, 2018.

[202] M. Morise, F. Yokomori, and K. Ozawa. World: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, 99(7):1877–1884, 2016.

[203] E. Nachmani, A. Polyak, Y. Taigman, and L. Wolf. Fitting new speakers based on a short untranscribed sample. In *International Conference on Machine Learning*, pages 3680–3688, 2018.

[204] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in Neural Information Processing Systems*, pages 4480–4490, 2018.

[205] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou. Neural voice cloning with a few samples. In *Advances in Neural Information Processing Systems*, pages 10019–10029, 2018.

[206] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5206–5210. IEEE, 2015.

[207] C. Veaux, J. Yamagishi, and K. MacDonald. CSTR VCTK corpus: English multi-speaker corpus for CSTR Voice Cloning Toolkit, 2017.

[208] M. J. Salganik. *Bit by bit: social research in the digital age*. Princeton University Press, 2017.

[209] F. Durand. Ethics and computational photography. `http://www.thecomputationalphotographer.com/wp-content/uploads/2019/05/25-Ethics-compressed.pdf`, MIT EECS & CSAIL, 2019. Accessed: 2019-06-14.