# Synthesising prosody with insufficient context

*Zack Hodari*


Doctor of Philosophy

Centre for Doctoral Training in Data Science

School of Informatics

University of Edinburgh

2022

# Abstract

Prosody is a key component in human spoken communication, signalling emotion, attitude, information structure, intention, and other communicative functions through perceived variation in intonation, loudness, timing, and voice quality. However, the prosody in text-to-speech (TTS) systems is often monotonous and adds no additional meaning to the text. Synthesising prosody is difficult for several reasons: I focus on three challenges. First, prosody is embedded in the speech signal, making it hard to model with machine learning. Second, there is no clear orthography for prosody, meaning it is underspecified in the input text and making it difficult to directly control. Third, and most importantly, prosody is determined by the context of a speech act, which TTS systems do not, and will never, have complete access to. Without the context, we cannot say if prosody is appropriate or inappropriate. Context is wide ranging, but state-of-the-art TTS acoustic models only have access to phonetic information and limited structural information. Unfortunately, most context is either difficult, expensive, or impossible to collect. Thus, fully specified prosodic context will never exist. Given there is insufficient context, prosody synthesis is a one-to-many generative task: it necessitates the ability to produce multiple renditions. To provide this ability, I propose methods for prosody control in TTS, using either explicit prosody features, such as $F_0$ and duration, or learnt prosody representations disentangled from the acoustics. I demonstrate that without control of the prosodic variability in speech, TTS will produce average prosody—i.e. flat and monotonous prosody.

This thesis explores different options for operating these control mechanisms. Random sampling of a learnt distribution of prosody produces more varied and realistic prosody. Alternatively, a human-in-the-loop can operate the control mechanism—using their intuition to choose appropriate prosody. To improve the effectiveness of human-driven control, I design two novel approaches to make control mechanisms more human interpretable. Finally, it is important to take advantage of additional context as it becomes available. I present a novel framework that can incorporate arbitrary additional context, and demonstrate my state-of-the-art context-aware model of prosody using a pre-trained and fine-tuned language model. This thesis demonstrates empirically that appropriate prosody can be synthesised with insufficient context by accounting for unexplained prosodic variation.

# Lay Summary

Text-to-speech (TTS) synthesis is the task of producing synthetic speech automatically from text. TTS has many applications, from smart assistants and voice user interfaces, to audiobooks and dubbing. However, the synthetic speech from these systems is not as expressive as natural human speech. In my research, I demonstrate that one reason for the lack of expressivity is the systems' inability to produce multiple versions of a sentence. As such, the first approach I develop is able to produce multiple renditions of a sentence.

Humans are able to perform a sentence in many different ways, possibly conveying different meanings. This is achieved through prosody: the use of pitch, loudness, and rhythm. Unfortunately, prosody has no clear written form. This means it is difficult to specify what prosody to use for a sentence. I develop new approaches that allow a human operator to control prosody more easily in TTS, providing control of emotion, attitude, and speaking style.

While a human-controlled TTS system is useful for the offline creation of media, such as audiobooks and dubbing, it is not suitable for real-time applications of TTS, such as smart assistants and voice user interfaces. In order to automatically predict better prosody we need to know what situation a sentence is being delivered within, for example: should the speech sound happy or sad, what was said in the previous sentence, or is a joke being told? This situational information is referred to as the context. In TTS, we have access to only a small amount of context and this is insufficient to predict prosody that is appropriate to a specific situation. The final approach I explore introduces new context information to automatically predict prosody. Using methods from natural language understanding, my approach is able to make a significant improvement over existing state-of-the-art TTS systems.

# Acknowledgements

I would like to thank my supervisor, Simon King. He has been a constant source of knowledge and advice. Over the years he has instilled in me the importance of forming a convincing narrative, both in research and teaching. Thank you for supporting me through each and every project. I'm also grateful to my advisors, Oliver Watts and Catherine Lai, whose advice was invaluable in shaping my ideas into practical and meaningful projects.

I owe a great deal to the many advisors, collaborators, and peers in Informatics. In particular, I felt privileged to be a part of CSTR, it is an abundant source of ideas and a repository of expertise—including knowledge of Edinburgh's best pubs. Thank you to all those who provided interesting discussion, helped improve my work, and read drafts of my papers and this thesis. During my PhD, I was lucky enough to pursue long internships. My development as a researcher is in no small part thanks to my mentors and peers at Google and Amazon. Thank you especially to Rob and Alexis. They, and many many others, influenced me in myriad ways and enriched my experience of becoming a productive researcher.

Finally, thank you to all the friends who made Edinburgh such a wonderful place. Both the CSTR and ILCC lunch groups were an excellent source of distraction, as were the bizarre tangents our office regularly found ourselves discussing. A massive thanks to Matt A and Bobby for bringing so many people together around board games. Thanks to John, Bobby, Matt, Matt, Matt, James, Tiffany, Carol, Jason, Jacob, and Cara for always being there for a coffee, a walk, a game, or a beer (or two). Thanks Ben, Sharon, Omar, Jan, and Sophia for making me love London. And thanks to Kaitlyn for many things, but especially for teaching me how to relax. Above all, thank you to my parents, I know I can always count on your unconditional love and support.

# Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Zack Hodari*)

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Speech is a rich form of communication that can convey information more efficiently than written language, especially social information, like emotion (Ben-David et al., 2016) and attitude (Mitchell and Ross, 2013). Speech conveys more than the utterance's written form; this additional information is communicated using prosody. However, unlike written language, prosody has no clear orthography. This leads to a challenge for text-to-speech (TTS) synthesis as we must produce prosody without any orthographic specification. My research focuses on controlling prosody without such an orthography of prosody: by learning representations of prosody, by investigating the interpretability of such representations, and by predicting prosody using the same contextual factors available to humans when planning prosody.

Speech consists of lexical content and prosodic content. **Lexical content** refers to the spoken words that make up an utterance, while **prosodic content** refers to the delivery of the lexical content. The choice of both lexical and prosodic content is imperative to communicating both effectively and efficiently.

**Text-to-speech** (TTS) synthesis is the process of generating a waveform from a sentence. In TTS, we take the lexical content as given, while we must choose an appropriate prosodic delivery for the fixed lexical content. Approaches that also generate or modify the lexical content—such as concept-to-speech (Taylor, 2009, Section 3.4), natural language generation (Gatt and Krahmer, 2018), and paraphrasing (Androutsopoulos and Malakasiotis, 2010)—are not considered here.

**Prosody** is the use of phrasing, timing, intonation, loudness, and voice quality to communicate: meaning, emphasis, emotion, humour, sarcasm, and other paralinguistic functions. Prosodic choices are influenced by many factors relating to the *context* in which a speech act occurs. For example, knowledge of what is new or known to your interlocutor may determine if you emphasise or reduce a word (Krifka, 2008). Or, your relationship with a person can change what emotion you express in your speech.

For a human speaker, the context of a speech act is readily available when making prosodic (and lexical) choices. However, in TTS, where we are given the lexical specification but must choose the prosodic delivery, there is often a lack of relevant prosodic context information. **Prosodic context** is the contextual information typically used by human speakers, consciously or otherwise, to plan their prosodic delivery. The relationship of prosodic context with other context is discussed further in Chapter 2. The lack of prosodic context in TTS leads to prosody being treated as unpredictable variation.[1] Thus, TTS voices that lack sufficient context produce speech that is not *appropriate* to the context, potentially leading to an "uncanny valley" phenomenon (Mori et al., 2012). In practice, TTS voices in consumer products are designed to take little risk with prosodic choices (Wan et al., 2019), leading to uninformative prosody.

Prosodic **appropriateness** is a measure of how well a prosodic rendition aligns with listener expectations for the speech act's prosody, given its surrounding context (Campbell, 2007; Cole, 2015). This metric for prosodic quality is defined with respect to an utterance's context, meaning that evaluating appropriateness requires the use of context (Wagner et al., 2019; Clark et al., 2019). Since there are multiple valid prosodies in any situation, defining which is "better" is not straightforward. By instead measuring appropriateness, we focus on understanding which prosodies are more/less acceptable.

The context required to predict appropriate prosody, as a human would, is very broad. Prosodic context ranges from *local* information such as: syntax, semantics, focus, affect, and relevant or nearby speech acts, to more *global* information such as setting, personality, interpersonal factors, general knowledge

[1] More precisely, when there is insufficient context, prosody appears to be *unexplained* variation. However, due to modelling assumptions made by typical TTS approaches, realistic prosody is *unpredictable*: only the non-existent average prosody can be modelled, as explored in Chapter 3.

(e.g. current news events), world/semantic knowledge (e.g. that a chair is something you sit on), and social knowledge (e.g. how to behave around a superior). Prosody is also influenced by previous prosodic choices in relevant speech acts, i.e. prosody used in the surrounding utterances. Different types of context, and their relation to prosody are discussed further in Section 2.2.6.

Unfortunately, much of this information is difficult to codify, and some is not attainable or is impractical to collect. Consider the relationship between conversation partners: even a customer service interaction includes complex interpersonal behaviour, such as subtle clues about personality and the hierarchical relationship between the speakers. Or, consider the internal emotional state of a speaker: since we only observe their somatic emotional expression (Clynes, 1977; Picard and Picard, 1997), it may not be possible to know the cause for certain prosodic choices. In addition, certain context information that can be annotated may be prohibitively expensive to collect. More generally, Lewis (1979) posits that interlocutors utilise a mental representation of a conversation. By this theory alone, there is unattainable context information. This limitation—that much of the prosodic context used by humans cannot be explicitly collected—is illustrated by the vertical dotted line in Figure 1.1. While internal information held by interlocutors cannot be annotated, it may be possible for future machine learning methods to capture similar contextual representations. Such learnt representations would, in effect, move the vertical line of limited context to the right.

The relationship of achievable appropriateness with respect to the amount of prosodic context used is depicted by Figure 1.1. The lower triangle, labelled "accounted-for prosody", illustrates how much of the prosody can be predicted deterministically: the more context available the more appropriate the predicted prosody can be. In practice, TTS models cannot exist past the vertical dotted line representing limited context. While we are far from reaching this boundary, it will become infeasible to push for more types of context information, this line of reasoning leads us to the same situation as typical TTS: dealing with *unexplained* variation. Unexplained variation includes both deterministic variation lacking context (i.e. unaccounted-for variation) and random variation.

Not all prosodic choices are deterministic; there is stochasticity in human prosodic behaviour (Goodhue et al., 2016). That is, the same intent can be conveyed in multiple ways by the same person in the same situation. This is not to

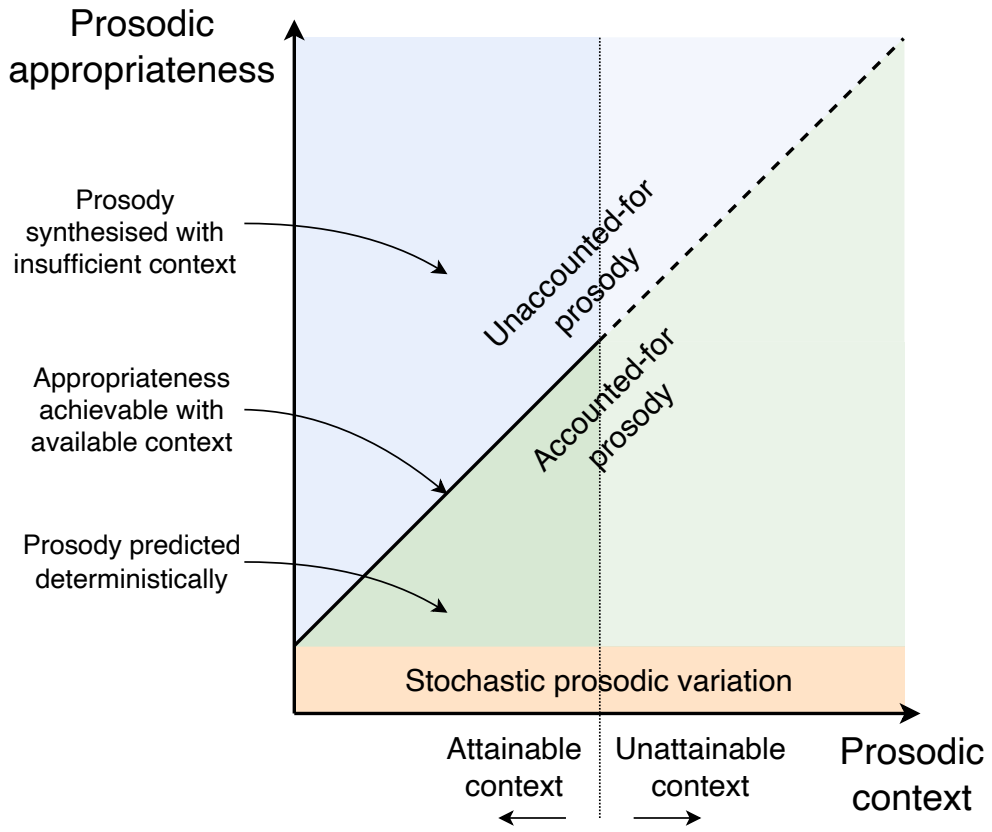



Figure 1.1: Illustration of the positive correlation between prosodic context and prosodic appropriateness, the limited range of attainable prosodic context, stochastic prosody that can be modelled regardless of context, and the resulting need for modelling unaccounted-for prosody with insufficient context. The relative sizes of categories in this illustration should not be interpreted as meaningful.

say that prosody can be modelled entirely as a random process—this would lead to inappropriate prosody—but to be fully natural, any model must consider the randomness present in some aspects of prosody. In Figure 1.1, "stochastic prosodic variation" represents this aspect of prosody. No matter what context is available, there is always some random behaviour that must be modelled. While stochastic variation is unexplained, it is separated from "unaccounted-for prosody" as there is no missing context.

To match human prosodic behaviour and improve appropriateness in speech synthesis, we need to synthesise the unaccounted-for prosodic variation, i.e. prosody with insufficient context. The approach taken in this thesis is to model the distribution of prosodic renditions an utterance could take on, turning prosody synthesis into a task of picking an appropriate rendition from all candidate renditions,

or a subset thereof. This selection task still requires knowledge of context, but we can rely on human-in-the-loop operators to intuitively pick an appropriate rendition, effectively inferring missing context.

By exploring new approaches to modelling prosody, I improve appropriateness across the three categories in Figure 1.1: incorporating additional context, correctly modelling stochastic variation, and exposing prosody control to human-in-the-loop operators. In Chapter 3, I address some problematic modelling assumptions, that are present in current TTS systems, in order to better capture the prosodic distribution. I design and expose interpretable control for human-in-the-loop operators in Chapters 4 and 5. In Chapter 5, I also investigate what context would be most beneficial to prosody prediction. And, in Chapter 6, I design a state-of-the-art TTS model that incorporates additional context. Summarising, I explore the following claim,

> *Appropriate prosody can be synthesised with insufficient context, but prosodic variation not determined by the available context must be controlled by a human or modelled probabilistically.*

## 1.1 Research themes

My research explores three themes:

1. **Prosody control** provides the ability to vary the prosodic delivery of an utterance.
2. **Interpretability** allows human users to intuitively understand the impact of the control inputs used to change the prosody.
3. **Appropriateness** is the degree to which prosody fits the context of a speech act.

I focus primarily on researching methods to control prosody, including learning representations of prosody—since prosody has no clear orthography. This approach splits prosody modelling into two tasks: control and prediction. Prediction can either be achieved using human-in-the-loop control or automatic context-based models.

Interpretability most directly aids human-in-the-loop control, making it much more efficient to interact with the control interface. Relying on a human provides

a solution (albeit not a scalable solution) to the insufficient context problem now, by exploiting human intuition (i.e. using inferred context). As additional context data becomes available we can transition from the labour-intensive human-in-the-loop design, to context-based control. Interpretability also benefits context-based prediction as it can be important to inspect and debug a system's behaviour; this is made much easier if there is human-interpretable meaning associated with the prosody representations.

### 1.1.1 Theme 1: Prosody control

In this thesis, **control** refers to the ability to vary the prosodic delivery of a sentence. Most TTS systems produce only a single prosodic rendition of a given sentence. As discussed in Chapter 3, this will lead to the production of average prosody which does not correspond to real human prosodic behaviour. As evidenced in that work, it is important that the range of prosodic choices is correctly modelled, this can include exposing choices through control inputs.

A controllable voice must have some mechanism to change the delivery, though the interface used can take any form. Control inputs include: human labelled concepts like emotion (Ekman, 1992; Fontaine et al., 2007) and attitude (De Moraes, 2011); prosodic correlates like $F_0$ (Fernandez et al., 2014; Wang, 2018), intensity (Wan et al., 2019; Klimkov et al., 2019), segment and pause duration (Turk et al., 2006; Rendel et al., 2017), and speaking rate (Henter et al., 2017b); engineered representations like ToBI (Silverman et al., 1992), SLAM (Obin et al., 2014), and wavelets (Ribeiro and Clark, 2015; Suni et al., 2015); or learnt representations (van den Oord et al., 2017; Wang et al., 2018a; Baevski et al., 2020). Discrete representations (Ronanki et al., 2016a; Wang et al., 2019b) can also be used to provide a different control interface that may be more usable or intuitive.

The level of detail in a control interface should be determined by the application. Detailed acoustic representations or learnt representations may be good for automatic context-based prediction, while human labelled concepts or discrete representations may be more suited to human-in-the-loop control.

### 1.1.2 Theme 2: Interpretability

Interpretability is a broad topic in machine learning, often focusing on explanations of a model's behaviour. Here, I use interpretability in a narrower sense: designing representations that can be intuitively understood by a human user. Generally, this is achieved by linking, directly or indirectly, to some more abstract human-understood concept through annotation.

By using a control input based directly on human labels, interpretable control can be achieved with the correct model design. However, this clearly incurs an annotation cost. Active learning (Settles, 2009) or pseudo-labelling (Lee, 2013) can be used instead to more efficiently collect labels, or re-use found resources. Representations like acoustic correlates of prosody, or those engineered correctly, can be interpretable by design.

Alternatively, representation learning can be augmented with approaches that make representations more interpretable, though making their interpretation reliable is an open challenge. Disentanglement can be used to remove unwanted information (Hsu et al., 2017b; Williams et al., 2021). These approaches can be noisy or imperfect, therefore, improving interpretability in learnt representations still requires human annotations—e.g. auxiliary tasks (Caruana, 1998; Ren et al., 2020). Another approach, is to assign meaning to unlabelled representations post-hoc using human annotation effort, this approach is well suited to discrete representations.

### 1.1.3 Theme 3: Appropriateness

Appropriateness is a measure of how well the prosody fits its context. It's important that the context of an utterance is considered when evaluating prosody, since it's only with respect to the context that prosody can be considered better or worse.

Currently, much of the TTS literature focuses on improving naturalness—a general term that includes all aspects of the quality of speech, including prosody (van Heuven et al., 1995, Section 1.4.6). Evaluating prosody quality separately from general measures of naturalness or listener preference is an open problem (Clark et al., 2019). Evaluating prosody quality together with acoustic and pronunciation quality leads to lower precision, as listeners must consider multiple

phenomena. This is especially important with advances in sequence-to-sequence acoustic modelling and neural vocoding, where naturalness tests misleadingly report that synthetic speech has reached parity with human speech (Kalchbrenner et al., 2018; Elias et al., 2021).

Fortunately, as the field shifts to focus on controlling style (Wang et al., 2018a) and prosody (Wan et al., 2019), there has been an increasing focus on prosody evaluation in the TTS literature (Latorre et al., 2014; Wagner et al., 2019), including the importance of context (Mendelson and Aylett, 2017; Loupi, 2017; Clark et al., 2019).

In addition to evaluation, this theme includes methods to generate more appropriate prosody. This could be achieved through new context features (Dall et al., 2016; Tyagi et al., 2020; Karlapati et al., 2021), architectures (Yu et al., 2019), losses (Ren et al., 2020; Williams et al., 2021), or data (Goodhue et al., 2016; Zen et al., 2019). As illustrated in Figure 1.1, increasing the amount of available context allows for more prosody to be modelled deterministically. Improved modelling of stochastic variation and the ability of prosody control to expose an appropriate rendition also form part of this theme.

### 1.1.4 Thesis outline

I focus primarily on one theme in each of the research chapters, the themes touched on in each chapter are summarised in Table 1.1. Chapter 3 focuses exclusively on control, investigating the issues that emerge when unexplained prosody is not modelled. Through this work, I demonstrate the need to factor prosody out from the phonetic content of speech, allowing it to be controlled explicitly when there is insufficient context.

Chapters 4 and 5 focus on interpretability using two different approaches to explore interpretable control. In Chapter 4, the motivation is to design a system usable for human-in-the-loop control; this is achieved using human annotations derived from found data. Chapter 5 uses discrete representation learning and is motivated by the need for a better understanding of what perceived effects unsupervised representations capture and what context would be most impactful to improve appropriateness. In these chapters, I attempt to control unexplained prosodic variation, and use human-in-the-loop control to operate the controllable

Table 1.1: Research themes explored in context chapters. Larger ticks indicate the primary focus of each chapter.

| | **Theme 1** Controllability | **Theme 2** Interpretability | **Theme 3** Appropriateness |
|---|---|---|---|
| Chapter 3 | ✓ | | |
| Chapter 4 | ✓ | ✓ | |
| Chapter 5 | ✓ | ✓ | |
| Chapter 6 | ✓ | | ✓ |

voices during evaluation. While no new context information is incorporated in these two chapters, the methods described in them could be extended to automatically predict prosody, as explored in Chapter 6.

I investigate appropriateness in Chapter 6, utilising additional contextual information to predict prosody. The proposed model uses a prosodically-informed loss to ensure the available context information is used to predict prosody. This model achieves state-of-the-art performance using contextualised word embeddings to encode semantic and syntactic context. Additional context can be easily incorporated into this system. While the prosodic context explored in Chapter 6 is far from exhaustive, my approach enables incremental improvement in prosodic appropriateness.

## 1.2 Contributions

The four chapters outlined above make up the research content in this thesis. The work in them has been published in the following peer-reviewed conferences:

Chapter 3 — *Using generative modelling to produce varied intonation for speech synthesis* (Hodari et al., 2019) presented at the Speech Synthesis Workshop 2019, Vienna, Austria.

Chapter 4 — *Learning interpretable control dimensions for speech synthesis by using external data* (Hodari et al., 2018) presented at Interspeech 2018, Hyderabad, India.

Chapter 5 — *Perception of prosodic variation for speech synthesis using an un-*

*supervised discrete representation of F0* (Hodari et al., 2020) presented at Speech Prosody 2020, Tokyo, Japan.

Chapter 6 — *CAMP: A two-stage approach to modelling prosody in context* (Hodari et al., 2021) presented at ICASSP 2021, Toronto, Canada.

The contributions and findings from these chapters are spread across the three research themes. For clarity, the key takeaways from each piece of research are broken down below by research theme:

**Theme 1** Controllability

Chapter 3 — By learning a distribution of $F_0$ using a variational autoencoder, I was able to develop a *synthesis-time approach to generate more varied prosodic renditions.*

Chapter 4 — Using pseudo-labelling, I proposed an approach to automatically label TTS data. This was used to *train controllable TTS voices without the need for human annotations.*

Chapter 5 — Using a multi-modal prior, I proposed a *novel latent variable model for sequence data.* In my model, for each phrase in the utterance, a multi-modal latent variable captured intonation and timing. This is important due to my earlier findings regarding average prosody and the need to consider the multi-modal nature of prosody.

By taking each mode as a category of prosodic variation, this model allows for *control using a learnt "orthography" of prosody.* However, the interpretability is not a given, and was also evaluated.

Chapter 6 — The proposed model *learns a word-level prosody representation from the spectrogram.* The representation is disentangled from phonetic content using two information bottlenecks. This model is intended for context-based control as it is not designed to be interpretable.

**Theme 2** Interpretability

Chapter 4 — The controllable voice uses emotion categories as the control input. I conducted a listening test to validate if the voice can successfully control perceived emotion using the control inputs. By comparing

the results with related work on human agreement for emotion labelling, I demonstrated that *the control inputs are interpretable.*

Chapter 5 — The discrete "orthography" learnt by the multi-modal prior is not guaranteed to be interpretable. I conducted qualitative interviews to understand if the discrete categories corresponded to certain prosodic behaviours. Unfortunately, no consistent behaviour was observed, possibly due to stimuli design and the small size of the experiment. However, it was clear that *different perceived prosodic behaviours were produced by the learnt categories.*

**Theme 3** Appropriateness

Chapter 3 — The phenomenon known as "average prosody" is where typical statistical models produce an average of the prosodic behaviour seen in the data. This average does not correspond to realistic prosody. While average prosody can be perceived in many TTS voices, it had not been formally studied. I designed an evaluation that *detected the flatter intonation of average prosody*, demonstrating that it is present.

In the same evaluation, I showed that *typical modelling assumptions in TTS voices are responsible for average prosody.* Many models implicitly generate from the mean of the prosodic distribution. However, much of prosody is multi-modal: it is due to this mismatch that typical models produce average prosody. By avoiding the mean, more realistic and varied prosody can be generated, i.e. stochastic variation of prosody is more effectively captured.

Chapter 4 — Different approaches for controlling the emotive voice in long-form content were compared. A listening test demonstrated that *prosodic variation must be appropriate to the prosody in the previous utterances.*

Chapter 5 — Through a qualitative evaluation of the discrete categories, it was clear that *adding context information relating to affect would be most useful to improving appropriateness.* This is likely specific to the children's audiobook style of speech that was used to train the model.

In addition, it was found that when listening to an out-of-context utterance with multiple distinct prosodic renditions, *listeners imagined contexts that would be appropriate to the different renditions.*

Chapter 6 — I proposed using a prosodically-relevant loss, instead of the typical spectrogram losses, when adding context information. This should ensure context is used to predict prosody, instead of focussing on frame-level acoustic quality. Related work in the literature used the same context features, but did not see an improvement in appropriateness. This suggests that *a prosodically-relevant loss is important when incorporating context information.*

Adding semantic and syntactic context features led to a very substantial improvement in quality, measured by listener preference. Compared to a strong state-of-the-art baseline, *my approach is 26% closer to reaching parity with natural human speech.*

### 1.2.1 Additional contributions

During the course of my research I produced two open-source software libraries: Morgana (Hodari, 2020a), and tts-data-tools (Hodari, 2020b). The research in Chapters 3 and 5 used these two libraries. Similar to the popular TTS toolkit, Merlin (Wu et al., 2016b), Morgana aims to make reproducing TTS models easy. Between these libraries, model training is handled by Morgana, while data preparation is separated out into tts-data-tools. This allows for more transparency of what is required to train a TTS voice. Morgana is a "model-first" wrapper of PyTorch (Paszke et al., 2017), and it provides support code for training, metric logging, data loading, and visualisation. Model-first describes the user-interface of the software; a user should only need to define a model and run that as an executable. Morgana's object-oriented design allows any process to be customised easily. tts-data-tools is a collection of tools and wrappers of other software, allowing for dataset-level pre-processing required for training neural-network-based TTS voices. In Chapter 4, I use the ModNN library, a piece of software written during my Masters by Research (Hodari, 2017a). While completing the research for Chapter 4, I contributed additional features to ModNN.

The four papers described, and their corresponding chapters, cover the bulk

of the work completed during my thesis. However, I also completed some smaller projects and collaborated on two other papers:

- In late 2017, I collaborated on the 2018 Voice Conversion Challenge (Lorenzo-Trueba et al., 2018) with Srikanth Ronanki, Sam Ribeiro, Felipe Espic, and Cassia Valentini-Botinhao. Voice conversion aims to convert an utterance (in a source speaker's voice) to sound like a given target speaker. Some training/adaptation data for the target speaker is used to define the voice identity.

  My contribution was aligning and improving the parallel data. I performed per-speaker HMM forced-alignment (Toledano et al., 2003) on the 3 datasets we were using. For parallel voice conversion, the two utterances must be the same length, initially we simply clipped the longer utterance. To avoid this loss of information, I also performed DTW alignment on the parallel sentences to upsample the shorter utterance.

  When bringing together the contributions of the team, our system did not provide a significant enough improvement to be competitive in the challenge.

- I experimented with WaveNet (van den Oord et al., 2016) in early 2018, both for TTS and neural vocoding. My motivation was to explore the ability to train with smaller found datasets, and to quickly create usable vocoders with smaller architectures for faster prototyping.

  Getting the vocoder stable with the CMU Arctic database (Kominek and Black, 2004) was relatively straightforward.[2] I found training for roughly 24 hours was required to create a neural vocoder with good enough quality to evaluate the pronunciation and prosodic quality of an acoustic model. Early in training the vocoder would struggle to produce consonants, especially plosives. To get satisfactory acoustic quality, the model needed 2-3 days of training.[3]

- I collaborated on Fong et al.'s (2019) paper: *Investigating the robustness of sequence-to-sequence text-to-speech models to imperfectly-transcribed training data*, presented at Interspeech 2019, Graz, Austria.

[2] After some difficulty with one implementation, I found more success with Ryuichi Yamamoto's implementation, available here: r9y9.github.io/wavenet_vocoder.

[3] A brief overview of these findings as well as audio samples can be found at the bottom of this tutorial: zackhodari.github.io/wavenet_tutorial.

- I collaborated on Karlapati et al.'s (2021) paper: *Prosodic representation learning and contextual sampling for neural text-to-speech* presented at ICASSP 2021, Toronto, Canada.

# Chapter 2

# Background

This thesis approaches the challenges of prosody modelling from a machine learning perspective. However, many of the ideas explored are motivated by existing linguistic and intonational phonology research. The contributions in this thesis are primarily related to the practical aspects of prosody modelling for speech synthesis, as opposed to a more general theoretical understanding of prosody. I begin by discussing text-to-speech technology and techniques in Section 2.1, before providing an overview of prosody and prosody modelling in Section 2.2. In Sections 2.3 and 2.4, I cover considerations and challenges relating to evaluation and data collection—including the data used in this thesis. In Section 2.5, I provide some machine learning background as a primer for the methods that are used and developed in this thesis.

## 2.1 Text-to-speech

Text-to-speech (TTS) synthesis is the process of rendering an audio waveform for a given sentence. TTS is typically split into two stages: the front-end and the back-end. The front-end handles text analysis, while the back-end handles acoustic modelling (Section 2.1.1) and vocoding (Section 2.1.2).

### TTS front-end

The front-end's main goal is to determine how to pronounce the input sentence as a sequence of words (Sproat, 2008). The front-end must perform, at a bare

minimum, text normalisation and grapheme-to-phoneme conversion.[1] It is also common to extract additional syntactic information (Black et al., 1998), however in more recent models this is not typically used, as human-defined front-end features are outperformed by a learnt linguistic encoder (Watts et al., 2019).

Text normalisation is the process of converting written language to spoken language (Ebden and Sproat, 2015). Written language may include non-standard words such as: abbreviations, dates, and numbers (Sproat et al., 2001).

Grapheme-to-phoneme conversion refers to the prediction of pronunciation from a sequence of graphemes. This process may use a pronunciation lexicon (Fitt and Isard, 1999), letter-to-sound rules (Taylor, 2009), a grapheme-to-phoneme model (Bisani and Ney, 2008), or a combination thereof.

### TTS back-end

The back-end must produce a waveform according to the phonetic specification provided by the front-end. In the past, TTS used non-parametric methods where a waveform was concatenated together from recorded speech units. Diphone synthesis (Hamon et al., 1989) was an early concatenative method, but this was superseded by unit-selection (Hunt and Black, 1996) and later by hybrid unit-selection (Kominek and Black, 2006). Hybrid unit-selection combines parametric speech models, introduced below, with the non-parametric unit-selection approach. It was state-of-the-art until 2016 (Zen et al., 2016) when neural vocoders advanced enough to create purely synthetic speech of higher quality than concatenated speech (van den Oord et al., 2016; Wu et al., 2019).

Parametric modelling of speech poses many challenges—concatenative methods avoided a number of these by design. There is a severe mismatch in length, information density, and information content between the back-end's input phonetic information and output waveform samples. Phonetic information has much lower information density than the waveform: 39 bits/s (Coupé et al., 2019).[2] In state-of-the-art TTS, the waveform is typically sampled at 24 kHz with a bit depth of 16 (Hsu et al., 2019; Merritt et al., 2018; Wu et al., 2020a). This means

[1] For character-based models, grapheme-to-phoneme conversion is performed implicitly by the acoustic model and is not part of the front-end (Wang et al., 2017b).

[2] Coupé et al. (2019) also report the information density in terms of speaking rate: the number of tokens spoken per second. They found that, across languages, speech contains an average of 6.6 syllables per second.

a parametric back-end must predict 24,000 16-bit amplitude values for every second of synthesised audio. Phones are a very compact representation of speech, and while the waveform does contain additional information and a lot of redundant information, this compactness is evident in the length mismatch and the difference in information density. This makes it challenging to train a model that maps from a phone sequence to waveform samples. Much of the information in the waveform is not represented in the phone sequence; this includes articulation, speaker identity, prosody, and channel information. The back-end must generate all information not present in its inputs.

To make parametric modelling of speech more tractable, the back-end is split into an acoustic model and a vocoder. In some cases the front-end provides additional inputs for the acoustic model, such as: part of speech, phrase breaks, or emotion labels (Black et al., 1998). Additional models can be incorporated into the back-end, such as intonation models (Wang, 2018). The acoustic model predicts an intermediate acoustic representation. This representation is extracted from the waveform using signal processing by the vocoder's analysis stage (Imai, 1983; Kawahara, 2006; Morise et al., 2016). Vocoders also convert this intermediate representation back to a waveform in their synthesis stage. Vocoder synthesis uses either signal processing (Imai, 1983; Kawahara, 2006; Morise et al., 2016) or, more recently, neural networks (Shen et al., 2018).

Acoustic modelling initially used hidden Markov models (HMMs) and decision trees to model acoustic features and phone durations (Tokuda et al., 1995; Zen et al., 2009). This was later improved by using two neural networks (NNs) to model acoustic features and phone durations (Zen et al., 2013) and, more recently, by using a single sequence-to-sequence model (Sotelo et al., 2017; Wang et al., 2017b). Thanks to neural vocoders (van den Oord et al., 2016; Shen et al., 2018), parametric models have greatly surpassed previous methods in terms of naturalness (Wu et al., 2019). Synthesis can also be performed directly from a phone sequence instead of using a separate acoustic model and vocoder (Weiss et al., 2020). Various approaches for NN-based TTS back-ends are illustrated in Figure 2.1. In the following two sections I discuss acoustic modelling and vocoding in more detail.

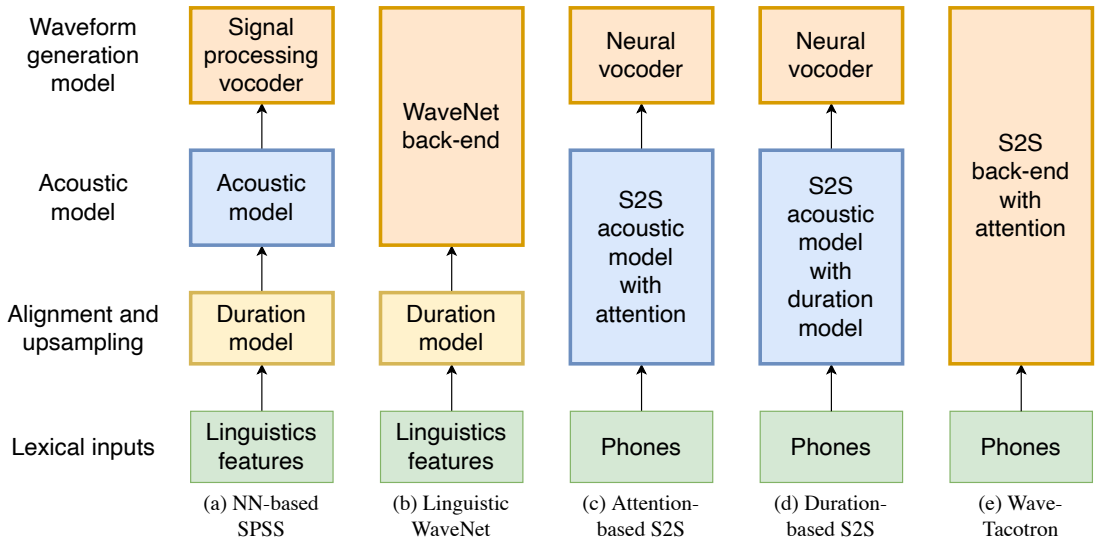



Figure 2.1: Different TTS paradigms, from SPSS to state-of-the-art S2S models. (a) NN-based SPSS with a traditional vocoder (Zen et al., 2013). (b) Linguistic WaveNet directly generating the waveform from linguistic features with a duration model (van den Oord et al., 2016). (c) S2S model with attention and a neural vocoder (Shen et al., 2018). (d) Non-attentive S2S model with a duration model and a neural vocoder (Yu et al., 2019). (e) S2S model with attention directly generating the waveform from phones (Weiss et al., 2020).

### 2.1.1 Acoustic modelling

I use the two most recent acoustic modelling techniques in this thesis: NN-based statistical parametric speech synthesis (SPSS), and sequence-to-sequence (S2S) models. SPSS uses separate duration and acoustic models (Figure 2.1a), whereas S2S models typically use attention to enable prediction of acoustic features directly from phonetic information in a single model (Figure 2.1c). However, there are many variations upon both paradigms, and in reality SPSS and S2S models bear a lot of similarity.

#### 2.1.1.1 Statistical parametric speech synthesis

NN-based SPSS uses two stages to predict the vocoder's acoustic features, as illustrated in Figure 2.1a (Zen et al., 2013; Wu et al., 2016b): a duration model and an acoustic model. SPSS models use linguistic features as input to synthesise speech. The duration model takes the linguistic features as input and predicts the duration of each phone. The linguistic features are upsampled to frame-level

using the predicted durations. At training time, natural durations, extracted using forced alignment (Toledano et al., 2003), are used to upsample the linguistic features. The acoustic model uses the upsampled frame-level linguistic features to predict acoustic features for each frame. The target acoustic features depend on the vocoder. However, in SPSS it is common to use mel-cepstral coefficients, band aperiodicity, $F_0$, and voiced-unvoiced state (Kawahara, 2006; Morise et al., 2016).

NN-based SPSS uses linguistic features as input to the duration and acoustic models, these were inherited from unit-selection and HMM-based SPSS. The linguistic features describe the sequence of phones to be synthesised. Other hand-engineered features are common, such as: tri-phones or quin-phones, part of speech, word class, pitch height, and a variety of structural information (Zen, 2006). A full overview of the linguistic features used in this thesis can be found in Appendix A. A majority of the features are a flattened representation of the utterance's structure—this means useful relational information is either poorly represented or discarded.

Modelling duration and acoustics separately is a limitation for NN-based SPSS, but it is a practical solution for handling mismatched input and output sequence lengths. This mismatch makes it difficult to define a differentiable loss. For NN-based SPSS, duration modelling is separate as backpropagating through the upsampling operation was non-trivial with auto-differentiation libraries available when NN-based SPSS was introduced. A few recent S2S models—including my research detailed in Chapter 6—use separate duration and acoustic models that are trained jointly, as illustrated in Figure 2.1d (Ren et al., 2020; Łańcucki, 2021), this bears similarity to NN-based SPSS models.

#### 2.1.1.2 Sequence-to-sequence TTS

Sequence-to-sequence (S2S) models provided a big improvement in naturalness for TTS, surpassing both NN-based SPSS and hybrid unit-selection (Wu et al., 2019). S2S modelling originated in NLP, which faces the same challenge of handling mismatched input and output sequence lengths. Sutskever et al. (2014) proposed S2S modelling for machine translation to resolve this issue. S2S models consist of an encoder and decoder, each operating on a sequence of a different length. Initially, S2S encoders used recurrent layers to summarise the input sequence in

one fixed-length representation (Graves, 2013). This context representation was used by the decoder to generate the output. Bahdanau et al. (2014) replaced this fixed-length representation with the attention mechanism.

Attention was developed to improve the transmission of information between the encoder and decoder in machine translation (Bahdanau et al., 2014). For each output time-step, attention produces a context vector that summarises the input information relevant to that output. Thus, attention aligns information in the input to the decoder output sequence. The ability to dynamically align information was particularly important for machine translation, where the source and target sentence can have a non-monotonic alignment. Attention is described in more detail in Section 2.5.1.4.

Typical S2S TTS models (Figure 2.1c) use a phonetic encoder which takes phone identity as input, instead of the linguistic features used in SPSS. The phone embeddings output by the encoder are attended over by an autoregressive acoustic decoder which outputs mel spectrograms for a neural vocoder (Shen et al., 2018). However, there are many variations on this architecture. Attention can be replaced with a duration model, as shown in Figure 2.1d. A reference encoder can be used to learn control features (Skerry-Ryan et al., 2018), making the S2S model an autoencoder. The decoder can predict the waveform directly (Figure 2.1e), instead of predicting spectrogram features (Weiss et al., 2020). Parallel decoders can be used to speed up the slow synthesis speed of typical autoregressive decoders (Li et al., 2019; Elias et al., 2020), using convolutional or self-attention layers (Tachibana et al., 2018; Li et al., 2019).

The attention mechanism has been improved upon in many forms, mostly to enhance robustness and stability (Chorowski et al., 2015; Battenberg et al., 2020). There even exist attention variants intended for specific applications, such as interpretability (Wang et al., 2019c), controllability (He et al., 2019), and summarisation (Gu et al., 2016). For TTS, we do not need to perform non-monotonic alignment like in machine translation. Therefore, different forms of attention have been proposed that parameterise monotonic alignments: alignments that do not progress backwards (Chiu and Raffel, 2018). Additionally, some approaches replace attention's alignment with durations either implicitly through hard alignments (He et al., 2019; Yasuda et al., 2019), or explicitly through a duration model (Yu et al., 2019; Ren et al., 2020). This suggests that attention is not

necessary for state-of-the-art S2S TTS, as observed in Chapter 6 and in the shift towards explicit duration models in the literature (Yu et al., 2019; Ren et al., 2020; Shen et al., 2020; Łańcucki, 2021).

While there are many minor differences between variants of these two paradigms, S2S has a few common design differences compared to SPSS:

- **Phone encoder** — S2S models use a phone encoder to learn rich representations from phone identity, instead of using human-defined linguistic features.
- **Joint duration modelling** — S2S maps between phonetic inputs and acoustic outputs directly in a single jointly-trained model, instead of using separately trained duration and acoustic models.
- **Neural vocoding** — S2S models predict mel-scale spectrograms and use neural vocoders, whereas SPSS typically uses signal processing vocoders.

However, these are design choices and can be modified in any SPSS or S2S model. The line between the two paradigms is a blurred one. If an S2S model used additional linguistic features, a duration model, a signal processing vocoder, or any combination thereof, would it become an SPSS model? While the classification of a given system as SPSS or S2S could be discussed at length, this is not important here; it is more important to acknowledge that the two paradigms are not so different.

### 2.1.2 Vocoding

Vocoders define a representation of the waveform that is easier for acoustic models to predict. Specifically, vocoding consists of two stages: analysis and synthesis. A vocoder's analysis stage converts speech waveforms to acoustic features—a more compact representation of the waveform. A vocoder's synthesis stage converts these acoustic features back into a waveform. The ideal synthesis stage should be agnostic of where the acoustic features come from: human speech or an acoustic model. It should also be robust to different speakers, accents, and languages.

Typical signal processing vocoders adopt the source-filter model of speech production. The source-filter model assumes that the source—e.g. vibration or frication of the vocal folds—is filtered into speech sounds by the shape of the vocal

tract and articulators (Stevens, 2000). These vocoders model the excitation signal (i.e. the source), aperiodic energy (for unvoiced sounds), and spectral envelope (i.e. the filter) as separate acoustic features (Kawahara, 2006; Morise et al., 2016).

To improve vocoder synthesis, it is important to consider the analysis stage. This can include extracting more compact acoustic features which can improve acoustic modelling. For example, we can use warped frequency scales, such as the mel scale (Stevens et al., 1937) and bark scale (Zwicker, 1961), since human perception of frequency is non-linear. This makes the acoustic features more compact, and thus easier to predict, without sacrificing quality. Alternatively, we can use additional acoustic features, such as phase, to improve synthesis using additional information (Espic et al., 2017).

More recently, neural vocoders have become widespread (Zhou et al., 2020). Neural vocoders vastly improve vocoder synthesis. They use a very simple analysis stage, typically extracting the mel-scale magnitude spectrogram. The first iteration of a neural vocoder was WaveNet, this used stacked dilated convolutions and residual connections (van den Oord et al., 2016)—however, this WaveNet performed acoustic modelling in addition to vocoding (Figure 2.1b). Tacotron-2 (Figure 2.1c) was the first application of WaveNet's techniques to the task of vocoding (Shen et al., 2018). The quality of WaveNet was big improvement on what was possible with traditional signal processing vocoders. Precisely why is not entirely clear, however it may be due to: increased receptive field from dilated convolutions, or the summarisation from residuals (van den Oord et al., 2016); larger, or higher quality data (Podsiadło and Ungureanu, 2018); or improved training techniques (Kalchbrenner et al., 2018).

While WaveNet's quality is excellent, the model is autoregressive and very slow at synthesis time. Many methods have been developed to parallelise neural vocoders (van den Oord et al., 2018; Prenger et al., 2019; Yamamoto et al., 2019) or make them more efficient (Kalchbrenner et al., 2018; Jin et al., 2018; Valin and Skoglund, 2019). Research on neural vocoders has also been influenced by source-filter theory (Wang et al., 2019a). Source-filter models can be very efficient, and have been used for signal compression. The most notable example is linear predictive coding (LPC) (Atal and Hanauer, 1971). The ideas from LPC have also been adapted and incorporated into neural vocoder architectures by predicting source and filter components (Juvela et al., 2019) or by predict-

ing linear prediction coefficients directly (Valin and Skoglund, 2019). Valin and Skoglund's (2019) LPCNet outperforms other fast neural vocoders like WaveRNN (Kalchbrenner et al., 2018), but not autoregressive models like WaveNet.

### 2.1.3 Controllability in TTS

Providing control over various aspects of speech can be important for many applications of TTS. Control of speech is very broad, and can include changing the speaker identity (Jia et al., 2018); accent (Henter et al., 2018a); or prosody, e.g. speaking style (Wang et al., 2018a), emotion (Henter et al., 2017a), prominence (Malisz et al., 2017), phrase breaks (Rosenberg, 2010; Rendel et al., 2017; Klimkov et al., 2017), or intonation patterns (Zou et al., 2021). Learnt representations can be used to control many aspects of speech, including those that are not easily labelled. If interpretability is important, e.g. for human-in-the-loop control, methods such as disentanglement can be used to guide what the representations learn (Williams et al., 2021), this is discussed further in Section 2.2.5.

Control in TTS is most often achieved through auxiliary features (Dehak et al., 2011; Luong et al., 2017). Other approaches for speech control include model-based adaptation (Swietojanski and Renals, 2014) and feature-space normalisation (Neto et al., 1995). These have been explored in TTS for emotion control, speaking style control, and expressive speech synthesis (Yamagishi et al., 2004; Schröder, 2009; Barra-Chicote et al., 2010).

In SPSS, control techniques roughly fall into two categories: explicitly labelled control and latent control. Labelled control provides human-usable control, however the labels are typically labour-intensive and expensive to collect. It's also possible to automatically label data at the expense of accuracy (Rosenberg, 2010; Cai et al., 2020), as explored in Chapter 4. Without access to labels, unsupervised methods must determine which variation is salient. Despite this additional challenge, latent control can achieve similar results to supervised control (Henter et al., 2018b). The fact that latent control can perform similarly to supervised methods may be, in part, related to the low inter-annotator agreement of labels (Roy et al., 2017).

The same approaches can be, and have been, applied in the S2S paradigm for: emotion control (Henter et al., 2018b), style adaptation (Prateek et al.,

2019), and prosody control (Wan et al., 2019; Klimkov et al., 2019; Yu et al., 2019; Ren et al., 2020; Mohan et al., 2021). However, for S2S models the use of representation learning through a reference encoder has become a common approach. A reference encoder provides a mechanism for TTS models to learn a representation that controls unlabelled variation in speech (Wang et al., 2018a; Kang et al., 2021).

## 2.2 Prosody

Now we move onto prosody: the variation in speech used to communicate additional information. Prosody is a channel of communication carried in speech alongside lexical information (Shattuck-Hufnagel and Turk, 1996). Prosody can augment the lexical information (Wallbridge et al., 2021) and make speech easier to comprehend by: indicating information structure (Calhoun, 2010), resolving ambiguities (Tran, 2020), or grounding communication (Clark and Brennan, 1991). Alternatively, prosody can convey additional meaning: holding or yielding the floor (Gravano and Hirschberg, 2009), marking irony or humour (Bryant, 2011; Gironzetti, 2017), or expressing emotion (Ben-David et al., 2016). Prosody can be unique to each speaker, it can signal linguistic functions, convey attitude towards the content, and express emotional state (Monrad-Krohn, 1947).

Prosody is realised through perceived variation in suprasegmental aspects of speech, such as intonation, loudness, timing, and voice quality. Evidence suggests that prosody has discrete elements, often referred to as prosodic constructions (Ward, 2019). Prosody operates over many different levels, from micro-prosody below the segment domain to suprasegmental prosody at the syllable, intonational phrase, and utterance domains, forming a hierarchical structure (Nespor and Vogel, 2007).

Prosody is a natural part of verbal communication. But, unlike written communication, which has a clear orthography, prosody is not generally transcribed. Punctuation can indicate prosodic phrasing, but there are mismatches between grammatical punctuation and pausing. In writing, prosody can be conveyed by other means, such as, acting directions in a script, or narrative writing of character behaviour in a book. Alternatively, prosody can be improvised by a speaker, and then transcribed. However, existing descriptions of prosody, whether original or annotated, only describe a fraction of what is perceived.

As with written language, the structure and usage of prosody varies across languages. In this thesis, I focus exclusively on English. There are multiple annotation schemes, taxonomies, and grammars for English prosody (Silverman et al., 1992; Dilley and Brown, 2005; Cole et al., 2017; Goodhue et al., 2016; Ward, 2019; Steedman, 2014), however there is no agreed upon prosodic orthography and the perceptual validity of such a categorical system is still "uncertain" (Steedman, 2000, Section 3).

In this section, I cover different aspects of prosody: its use in communication (Section 2.2.1), how to annotate it (Section 2.2.2), its acoustic realisation (Section 2.2.3), how it is structured (Section 2.2.4), learning new representations of it (Section 2.2.5), and what information is used to plan it (Section 2.2.6). Bringing all this together, I discuss prosody modelling in Section 2.2.7.

### 2.2.1 Functions of prosody

Prosody serves many communicative purposes, covering both linguistic and paralinguistic functions. These are the goals used when planning prosody. In speech act theory (Searle, 1969), prosody can contribute to illocutionary force (Cole, 2015)—the intended meaning of a speech act (Austin, 1975). A statement may serve only to inform the interlocutor (1-a), while a question may aim to make them do something, i.e. share information (2-a). Illocutionary force also considers that speech acts can communicate other intentions: a statement may be a brag (1-b), or a question may be rhetorical (2-b).

(1) Statements

a. I went on **holiday**. (*inform*)

b. **I** went on holiday. (*brag*)

(2) Questions

a. Do you know what **time** it is? (*solicit information*)

b. Do **you know** what time it is? (*rhetorical*)

Prosody can play a role in grounding illocutionary force. For example, delivering an apology requires more than just a sequence of words—other requirements, including prosody, must be satisfied. As observed in Chapter 5 for synthetic speech, the same "apology" can be seen as sincere, insincere, or forced, depending on its delivery. However, the mapping between prosody and illocu-

tionary force is complicated, meaning that even with sufficient context, achieving the intended effect through prosody prediction or human-in-the-loop control is difficult.

Prosody can signal linguistic functions. Newness and givenness can be signalled through accent placement (Hirschberg and Pierrehumbert, 1986, Section 4.2). Prominence and pausing can be used to resolve semantic ambiguity (Cutler et al., 1997). Pragmatic information, such as the connotations of words and concepts—which vary according to a speaker's views—can be conveyed through speaking style or voice quality (Gobl and Chasaide, 2003), although these mechanisms are complicated (Barth-Weingarten et al., 2009).

Paralinguistic effects add more richness to verbal communication. Emotion, mood, humour, and irony are all illustrated using prosody, as well as through lexical choices and body language. Attitude—such as authoritative, friendly, or uncertain—is another paralinguistic function of prosody that can indicate information about a speaker's mental state, or their views towards a topic or interlocutor (De Moraes, 2011).

### 2.2.2 Prosodic annotation

To study the functions of prosody and to train models for prosody synthesis, it can be useful to have a compact representation of prosody. Unlike natural language which is codified both in written language and phonetics, prosody has no agreed upon prosodic orthography. As such, a lot of research effort has focused on codifying prosody. While I do not make use of these methods for annotating prosody, the hypotheses explored in Chapter 3 and my ideas for learning a discrete representation in Chapter 5 are inspired by these contributions.

Intonation, along with other aspects of prosody—including pausing, prominence, and voice quality—exhibits discrete structure. This has been studied extensively in intonational phonology (Pierrehumbert, 1980; Silverman et al., 1992; Hirschberg, 1999; Ladd, 2008; Cole, 2015). Goodhue et al. (2016) proposed an "intonational bestiary": a collection of prosodic renditions categorised according to a taxonomy of discrete intonation contours. Similarly, Ward (2019) discusses various prosodic constructions: discrete prosodic structures used to convey different information. As explored by Goodhue et al. (2016), three well-known

constructions include: (3-a) rise-fall-rise (Ward and Hirschberg, 1985; Wagner, 2012; Constant, 2012), (4-a) contradiction contour (Liberman and Sag, 1974; Ladd, 1980; Ward and Hirschberg, 1985), and (5-a) yes/no rise (Pierrehumbert and Hirschberg, 1990; Bartels, 2014; Truckenbrodt, 2011)

(3) Rise-fall-rise—example from Constant (2012)

A: Why isn't the coffee here?

a. B: I don't know. I was **expecting** there to be **coffee**...

(4) Contradiction contour—example from Goodhue and Wagner (2018)

A: Alvarado doesn't like movies.

a. B: **Alvarado** likes movies.

(5) Yes/no rise—example from Gravano et al. (2008b)

A: Marianna made some great marmalade.

a. B: Marianna made the **marmalade**?

B: I thought she was allergic.

However, attributing prosodic renditions to patterns in a taxonomy is not straightforward and requires expensive human annotation, not least because annotation is subject to variation in human perception (Roy et al., 2017). Ward (2014) explored the variation present in the Switchboard (Godfrey et al., 1992) and Maptask (Anderson et al., 1991) corpora using PCA, finding many prosodic forms with many different purposes. Categorising these is difficult, and determining what they are used for is even more so. Lai (2012a) found evidence that some speakers use the contradiction contour in a contradiction context, though other speakers did not behave as consistently. Goodhue et al. (2016) found some behaviours can be difficult to elicit or detect. In the appropriate context the incredulity contour was never observed in their data, or at least it wasn't perceived. Goodhue et al. (2016) discuss how this could be related to experimental design, or because the incredulity contour may not be a valid construction—it may be a variant of rise-fall-rise, as suggested by Hirschberg and Ward (1992). These issues demonstrate the difficulty of creating a taxonomy of prosodic forms and understanding their usage.

The classic approach to prosodic annotation is Tone and Break Indices (ToBI) (Silverman et al., 1992). The ToBI paradigm aims to describe intonation and phrasing through discrete categories, such as high and low tones. Timing is represented implicitly in ToBI's pitch accent and boundary annotations. ToBI

was developed from autosegmental-metrical theory which posits that intonation is made up of a sequence of distinct elements (Pierrehumbert, 1980). It captures the perceptual delivery of prosody, but requires expert annotators, and annotator agreement is low (Syrdal and McGory, 2000).

The more recent rapid prosody transcription (RPT) paradigm simplifies prosody annotation through word-level binary decisions and "wisdom of the crowd" (Cole et al., 2017). RPT has been demonstrated for prominence and phrase break annotation. Compared to annotation schemas like ToBI which codify prosody, RPT focuses on capturing prosodic variation as perceived by humans. Annotations from cohorts of less than 5 annotators are shown to contain a lot of variation, this makes them less reliable but also demonstrates the presence of different opinions. RPT's utility lies in its use of inter-annotator agreement statistics, which makes it possible to easily measure annotation consistency. Despite the use of more annotators than other paradigms, the simple task design means RPT is much lower cost and has better annotator agreement.

Alternatively, stylised parameterisations of prosody can be used to more precisely describe prosody. Stylisation is the process of identifying perceptually salient prosodic events and representing their characteristics parametrically, e.g. their shape. Representations such as Tilt (Taylor, 1998), ProsoGram (Mertens, 2004), and SLAM (Obin et al., 2014) provide parameterised representations that are useful for prosody synthesis. Despite the additional detail compared to, say, prosodic constructions, stylisation also risks omitting useful information (Obin, 2011, pp. 53) as they use small sets of human-defined parameters.

By design, ToBI, RPT, prosodic constructions, and stylised parameterisations omit detail present in the prosodic realisation. When these are used to control TTS systems, the acoustic model must generate the missing acoustic detail, including micro-prosody. The resulting prosody may deviate from what was expected, as the model must extrapolate from these high-level annotations. However, this is not a failure in the design of annotation schemas, it is the nature of TTS. Fundamentally, TTS is a generation task: it must add detail to an underspecified input. It would be more challenging to precisely and accurately specify all prosodic detail with these representations. That is, annotation schemas are useful because they omit the least important information.

### 2.2.3 Acoustic correlates

Acoustic correlates capture the objective acoustic realisation of prosody (Wagner and Watson, 2010), contrasting with annotation schemas which attempt to capture perceptual effects, such as prominence and phrasing. Acoustic correlates are extracted algorithmically from the waveform and can be used to control prosody in TTS. The acoustic correlates: $F_0$, intensity, segment and pause duration, and spectral tilt, are counterparts of perceptual correlates of prosody: intonation (or pitch), loudness, timing (or quantity), and voice quality.

The perceptual correlates of prosody relate to perceptual effects. For example, loudness and timing, and to a lesser extent intonation, correlate with prominence (Kochanski et al., 2005). By working with acoustic correlates, it is possible to control such perceptual effects, such as prominence (Malisz et al., 2017). While acoustic correlates are more detailed than annotations of prosody, there is a gap between acoustic correlates and the prosodic information listeners perceive in speech. Despite this, acoustic correlates are popular in part because they are cheaper and easier to extract than human annotations.

**Intonation** is the perceived fundamental frequency ($F_0$) at which the vocal folds vibrate. The signal produced by this vibration is often referred to as the source, or the excitation signal (Stevens, 2000). In a wideband spectrogram these glottal pulses—the source—can be seen as vertical pulses of energy. The distance between these pulses are a single pitch period. $F_0$ is defined as the reciprocal of a pitch period's duration. In a narrowband spectrogram, $F_0$ can be seen as harmonics, which present as stacked horizontal curves. Unvoiced sounds have an undefined $F_0$, but there is evidence that listeners perceive pitch during unvoiced segments (Taylor, 2009, Section 9.1). Many methods exist to automatically extract $F_0$ (Talkin, 2015; Morise et al., 2016; Kim et al., 2018).

**Loudness** is difficult to measure in digital recordings as amplitude values depend on the levels set by the recording engineer. Loudness' acoustic correlate, intensity, is the logarithm of the short-term average energy (Toledano et al., 2009, pp. 1286). Intensity can also be approximated using the first cepstral coefficient, $C_0$. Shimmer, another acoustic feature related to loudness, measures the relative variability of the signal's amplitude (Teixeira et al., 2013).

**Timing**, or the rhythm of speech, refers to: shortening or lengthening of

syllables, change of speaking rate, phrasing, pausing, and filled pauses (Turk et al., 2006). In TTS, we often model the acoustic realisation of timing, known as surface timing (Taylor, 2009, pp. 254). Surface timing is the durations of units in speech, often measured as the durations of segments (i.e. phones) and pauses. Durations are typically extracted using forced-alignment (Toledano et al., 2003). To account for errors in this automatic process, manual alignment can be performed by human annotators. Effects such as coarticulation mean that surface timing cannot be an entirely accurate measurement (Turk et al., 2006). Surface timing, like with other acoustic correlates, does not capture all perceptual effects. For example, phrase breaks that have no associated pause or only a baseline reset must be annotated manually (Cole et al., 2017). Non-silent pauses are also important, but can be much harder to extract reliably (Székely et al., 2019). In addition to duration, speaking rate—typically measured in syllables or words per second—can be modelled in TTS systems (Habib et al., 2020). The relationship between durations and speaking rate is non-linear and depends on segmental information, utterance length, and other contextual factors (Yuan et al., 2006).

**Voice quality**, or timbre, is often described as the colour or texture of speech (Truax, 1999). Voice quality can include behaviours such as breathy, creaky, whispered, or tense speech and can convey emotion, mood, and attitude (Gobl and Chasaide, 2003). Voice quality has wide-ranging effects on the speech signal, affecting both the source and filter in the source-filter model of speech production (Stevens, 2000). The broad impact of voice quality makes it difficult to measure. Additionally, it is rarely controlled in TTS as synthesis of different voice qualities is challenging. Many correlates of voice quality are based on spectral properties, such as the spectral envelope (Terasawa et al., 2005), which corresponds to the shape of the vocal tract, i.e. the filter. Voice qualities can impact the source by producing irregular source behaviour or removing voicing entirely—as with whispered speech. The acoustic correlate jitter captures irregular source behaviour. Jitter is the variation of the pitch period over time (Teixeira et al., 2013). Due to the wide-ranging impact of voice quality, defining new acoustic correlates using data is a promising approach. Kane and Gobl (2011) represent speech using wavelets and fit a linear model for each voice quality class, they find that the slopes of the linear models are useful discriminative features for the voice qualities investigated.

Together these acoustic features describe most of prosody, but do not completely define it. Fundamentally, there is gap between what is captured by these physical measures and what is perceived by listeners. Additionally, automatic extraction of acoustic correlates can be subject to extraction errors. For example, pitch tracking error modes include pitch halving, pitch doubling, and incorrect voicing detection, although improved methods, such as ensemble methods, can mitigate errors (Drugman et al., 2018). These limitations—completeness and accuracy—are part of the motivation for learning representations of prosody. Before moving on to discuss representation learning, it is important to introduce the concept of domain.

### 2.2.4 Prosodic domain

Prosody operates over different linguistic constituents. We do not need to consider this for acoustic correlates as they are represented in the time domain. However, in prosody modelling, considering which linguistic constituents to use in a model is important as it can impact downstream performance (Wang et al., 2019b). In linguistics, **domain** refers to the linguistic constituent over which an effect occurs.

Prosody has structure over multiple domains (Nespor and Vogel, 2007). Evidence from intonational phonology suggests that this hierarchical structure is recursive (Ladd, 1986). While multiple hierarchies are described in the literature (Ribeiro, 2018b, Section 3.1.2), a common version includes: syllable, prosodic foot, prosodic word, clitic group, phonological phrase, intonational phrase, and utterance (Selkirk, 1980). Some of these domains are difficult to work with in TTS as they require expensive annotation. Annotation introduces further challenges relating to the perception of boundaries within a domain. What is important for this thesis is that prosody can operate over all of these domains.

Utilising a prosodically-relevant domain is an important inductive bias for learning to model prosody, as demonstrated by Wan et al. (2019) and Wang et al. (2019b). Prosody also varies over longer domains, such as utterances and paragraphs, or turns and dialogues (Farrús et al., 2016). Intuitively, the variation expressed at these domains relates to higher level information, including: emotion, attitude, dialogue structure, or setting.

The syllable is the shortest domain that prosody operates within (Itô, 2018), carrying lexical stress as well as prosodic prominence. Although micro-prosody exists below the segment level, this should not need to be explicitly considered as it should be handled implicitly by the acoustic model.

The intonational phrase is the longest unit below the utterance (Pierrehumbert, 1980) and is another candidate for a prosodic domain. It is typically defined by the presence of perceived phrase breaks, which may or may not correspond to a pause or a baseline reset. Domains in the prosodic hierarchy do not necessarily align with syntactic structure (Selkirk, 1980). Thus, locating phrase breaks in speech is difficult to do algorithmically, and typically relies on human annotation (Silverman et al., 1992; Cole et al., 2017). Predicting the placement of phrase breaks at synthesis time is also very challenging. Despite pause placement being a core task in prosody synthesis, there is limited work in this area, as discussed in Section 2.2.7.

### 2.2.5 Prosodic representation learning

Representation learning provides another method to represent prosodic information. Unlike annotations and acoustic correlates, representation learning uses variation observed in data to determine which information is worth capturing. By learning from data, representation learning can capture any aspect of prosody; whereas for acoustic correlates new features can be difficult to design, e.g. for voice quality. Additionally, learnt representations do not require expensive human labelling like prosodic annotations. Learnt representations exist in an abstract embedding space, this structure is amenable to other downstream machine learning models.

Representation learning avoids certain pitfalls of manually defined prosody representations, such as unreliability in human annotations, incompleteness in stylised representations, and extraction errors in acoustic correlates. However, it has its own challenges. Most notably, learning to separate out, or disentangle, relevant information is difficult, and in some cases may not be theoretically possible. For example, micro-prosody is linked closely to segmental structure, and there are correlations between syntax and prosody (Köhn et al., 2018). Thus, representation learning models must be carefully designed to ensure the correct information is disentangled.

As with many machine learning methods, it is hard to interpret what a representation learning model captures. Therefore, evaluating what information is, or isn't, contained in representations is a common approach to validate learnt representations. Extrinsic evaluations like downstream tasks (Schölkopf et al., 2021, Section 2.5), or intrinsic metrics like mutual information (Schölkopf et al., 2021, Section 3.6) can be used to determine what the model has, or has not, learnt.

Representation learning models rely heavily on their loss. Self-supervised losses, such as autoencoder reconstruction (Skerry-Ryan et al., 2018) and contrastive learning (Baevski et al., 2020) are common in the representation learning literature (Wan et al., 2019; Devlin et al., 2019; Baevski et al., 2020, 2022). Autoencoders attempt to reconstruct the input following an information bottleneck. Contrastive learning, another self-supervised approach, defines a loss using positive and negative pairs of data points, such as predicting if an audio sample follows the current audio clip (Schneider et al., 2019). Self-supervised losses enable the design of powerful inductive biases that guide what information the model is likely to learn.

There are many techniques that can affect what a model learns. Auxiliary tasks, conditioning, and information bottlenecks, illustrated in Figure 2.2, are common components in representation learning models.

- **Auxiliary tasks** can be used to encourage certain information to be included in the representation (Figure 2.2a), and gradient reversal can be combined with auxiliary tasks to remove information from a representation (Figure 2.2b).
- **Conditioning** involves adding additional features as input to the model. This is typically information we don't want the representation to capture; by conditioning, it becomes redundant to represent this information in the learnt representation. For example, when learning a prosodic representation, conditioning on phonetic features can help disentangle the representations (Figure 2.2c).
- **Information bottlenecks** typically involve limiting the number of dimensions in the representation (Figure 2.2d). This approach is particularly effective when combined with conditioning or auxiliary tasks. The bottleneck

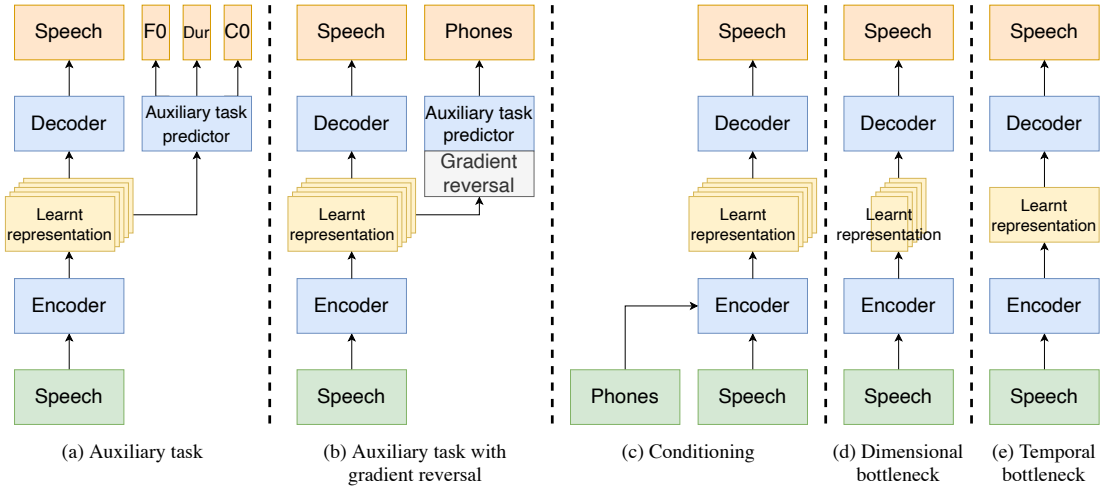



Figure 2.2: Disentanglement techniques in representation learning using an autoencoder. (a) Auxiliary tasks to encourage certain information to be captured. (b) Auxiliary task with gradient reversal to encourage disentanglement. (c) Conditioning to make phonetic information redundant. (d) Dimensional information bottleneck. (e) Temporal information bottleneck.

forces the model to capture the information most useful for the auxiliary task, as opposed to including all information. In the case of conditioning, a bottleneck makes it less likely for the representation to capture redundant information.

Information bottlenecks can also be temporal (Figure 2.2e). A temporal bottleneck can simply be at the utterance domain, as is common for controllable speech synthesis models (Skerry-Ryan et al., 2018; Wan et al., 2019; Karlapati et al., 2021). However, we can choose a domain for the temporal bottleneck that matches the domain of the information we want to represent. For example, using a phrase-domain temporal bottleneck means the representation will have a dynamic number of vectors: one for each phrase in an utterance. This is a powerful inductive bias for encouraging the model to represent information from the chosen domain (Wang et al., 2019b).

A common approach to unsupervised representation learning for controllable TTS uses autoencoders or variational autoencoders (VAE) (Kingma and Welling, 2013). VAEs have been applied to TTS (Hsu et al., 2019; Akuzawa et al., 2018), voice conversion (Hsu et al., 2017a), and prosody modelling (Wang et al., 2019b; Wan et al., 2019). Discrete representations have also been incorporated into the

VAE framework, which provides another information bottleneck (Rolfe, 2016; van den Oord et al., 2017). Prior work using VAEs has focused on modelling segmental features (van den Oord et al., 2017; Hsu et al., 2019), with some applications to prosody modelling, e.g. for prosody transfer (Wan et al., 2019) and learning unsupervised intonation representations (Wang et al., 2019b).

Many representation are designed to capture variation at the sentence domain not otherwise explained by the phonetic content (Skerry-Ryan et al., 2018). It is not clear exactly what information such representations capture, as they summarise over an arbitrary number of intonational phrases. Typically the variation captured is described as sentence style. The importance of a representation's domain and the perceived content of learnt representations is discussed in Chapter 5.

Before discussing prosody modelling in Sections 2.2.7, we now turn to context: the information missing from the input sentence in TTS.

### 2.2.6 Context

**Context** is the information used by humans to choose their words and prosody for a given speech act. Context in linguistics is hard to define formally, so definitions are often constructed specifically for individual problem domains (Goodwin et al., 1992). Context includes, but is not limited to: information structure, paralinguistic and social information, common ground, background knowledge, and setting—social and spatial information relating to a speech act (Goodwin et al., 1992, Chapters 2 and 3).

**Prosodic context** refers to the information used by humans when planning prosody. There is a wide range of prosodic behaviour influenced by context (Wagner and Watson, 2010). Prosodic context may be the same as "linguistic context", it may be a proper subset, or it be composed of additional information (Grice, 1989, Chapter 6). In this section, I detail relevant challenges surrounding context: it can be local or global, it is broad, it can influence prosodic behaviour (Section 2.2.6.1), and it is hard to collect (Section 2.2.6.2).

Different types of context information change at different rates, and affect prosody over different domains (Cole, 2015). I categorise context as local or global: local context varies throughout a dialogue or monologue, whereas global

context is typically static. However, the line between global context and local context can be blurred. For example, the setting of a speech act may be static, or may change over the course of a conversation. Similarly, local context such as emotion, which can change continuously, could stay the same for the duration of a conversation. Other types of local context (such as information structure) are inherently linked to individual utterances (Krifka, 2008), and will always vary.

#### 2.2.6.1 Relationship between context and prosody

Here, I demonstrate the breadth of prosodic context by providing specific examples of the relationship between context and prosody. Goodwin et al. (1992) illustrates the breadth of context by defining four wide-reaching categories of context: setting, behavioural environment, language as context, and extrasituational context. Similarly, Lewis's (1979) conversational scoreboard provides a framework for considering all contextual information held in speakers' minds that is relevant to a dialogue. While these formulations may include context information we cannot practically collect, they do illustrate how broad and complex context is.

Syntax and semantics, and information structure more generally (Chafe, 1976), can influence prominence placement to indicate focus or givenness (Krifka, 2008). Information structure can also resolve ambiguities surrounding relative salience (Lewis, 1979). The theme (topic of the utterance) and rheme (comments on the theme) may be used to convey new connotations or attitudes (Halliday and Matthiessen, 1999) and the prosody should reflect these according to the common ground (Tench, 2003).

Common ground is the information shared between dialogue partners (Stalnaker, 1974) or, in monologues, between the speaker and the audience. This can include background knowledge, information explicitly stated, or inferences presupposed by a speaker. Background knowledge is more global and covers multiple sources: general, encyclopedic, or common knowledge (e.g. current news events); semantic knowledge (e.g. that a chair is something you sit on); and social knowledge (e.g. how to behave around a superior). Background knowledge intuitively plays into prosodic appropriateness as a speaker will hold certain feelings about different knowledge which can manifest as emotions and attitudes.

Presuppositions are assumptions brought about implicitly in the course of natural language (Stalnaker, 1974). Prosody is used to indicate new presuppositions with less ambiguity, as exemplified by Krifka (2008),

(6) Presuppositions

a. John only showed Mary the **pictures**.

b. John only showed **Mary** the pictures.

(6-a) presupposes there were other things to show and they were not shown to Mary, while (6-b) presupposes there were other people and they were not shown the pictures.

Emotion and attitude both affect prosodic behaviour (Klabbers et al., 2007; De Moraes, 2011). Other paralinguistic context such as body language could equally provide useful information for determining appropriate prosody. Social information can impact prosody in dialogues and monologues. For example, interpersonal attitudes such as confidence, persuasion, sarcasm, and superiority are conveyed through prosody (Mitchell and Ross, 2013).

During dialogues, the discourse state can be signalled through prosody; e.g. terminal rises can suggest the discourse has not reached a stopping point (Lai, 2012b, Chapter 5). However, the mapping between context and prosody, or similarly between prosody and pragmatics, is not one-to-one. For example, in task-oriented dialogues, the rise-fall-rise construction can communicate a desire to coordinate and synchronise, or for a dominant interlocutor the same rise-fall-rise construction is used when eliciting confirmation (Lai, 2012b, Chapter 6).

Prosody planning in humans is a complex topic and the mechanisms used by humans are not considered in this thesis. Instead, I am concerned with modelling appropriate prosody in TTS. Cole (2015) reviews the myriad ways in which prosody signals context. Modelling improvements that incorporate wider context information would also provide an invaluable tool for computational linguists.

Even when context is available, incorporating it into TTS is non-trivial. Prevost and Steedman (1994) present a rule-based grammar for generating ToBI annotations based on a discourse model and a knowledge base. Efforts to utilise additional context in more recent systems have seen varying degrees of success (Dall et al., 2016; Ijima et al., 2017; Ribeiro et al., 2017; Rosenberg et al., 2018; Hayashi et al., 2019; Fang et al., 2019). Advances in S2S modelling (Tyagi et al.,

2020; Kenter et al., 2020) and representation learning (van den Oord et al., 2017; Wan et al., 2019; Wang et al., 2019b) have allowed prosody models to exploit context information more effectively, as we will see in Chapter 6.

#### 2.2.6.2 Annotation of context

Unfortunately, most contextual information is difficult to collect, requiring expensive human annotation. By definition, constructs such as Lewis's (1979) conversational scoreboard cannot be fully codified as they exist solely in the minds of dialogue participants. The same goes for information like emotion, which exists as an internal emotional state, and can only be inferred through external expression, i.e. through somatic expression (Clynes, 1977). Instead of attempting to collect the most complete and fundamental context information, it may seem reasonable to target more coarse context, such as the perceived external emotion, rather than underlying social dynamics. However, defining annotation schema for more coarse information can also be non-trivial, as is the case for emotion annotation (Douglas-Cowie et al., 2003).

Due to the wide range of context types that can influence prosodic behaviour, annotation for TTS data is prohibitively expensive, even if other practical challenges are ignored. Further, annotated context must be specified (or inferred) at synthesis time.

Automatic annotation of certain context information is possible, especially for low-level context such as syntax and semantics. The use of additional context derived automatically has been investigated for SPSS (Dall et al., 2016; Aubin et al., 2019). For S2S TTS—a relatively new paradigm—this has only recently been considered, and only using syntactic and semantic information (Hayashi et al., 2019; Fang et al., 2019; Kenter et al., 2020; Tyagi et al., 2020; Karlapati et al., 2021) (cf. Chapter 6).

Prosodic context includes different information depending on the mode of speech; choosing what context is most informative or relevant is, in itself, a challenge (cf. Chapter 5). For example, prosody in dialogues relies on more social information as it is used to establish common ground and rapport (Cassell et al., 2007) and will change as interlocutors converge (Sinha and Cassell, 2015). Whereas, prosody in monologues may be influenced more by the intended au-

dience (Montaño and Alías, 2017). Certain local context information, namely syntax and semantics, are likely to be useful for all modes of speech.

Together, these challenges, in particular that context is broad and hard to collect, mean we will always lack sufficient context needed to model appropriate prosody perfectly in TTS. While any additional context information manually annotated or automatically extracted should be helpful when synthesising prosody, we need to consider how to improve prosody with insufficient context, as posed by this thesis. My broad approach is to expose interpretable prosody control. When there is insufficient context, a human-in-the-loop can control the prosody. Alternatively, when relevant context information is available, control parameters can be predicted automatically.

### 2.2.7 Prosody modelling

Having discussed what prosody is used for, how it can be represented, and how context influences prosodic choices, I now cover methods to model and synthesise prosody. During synthesis in TTS systems, prosodic information is generated by the front-end and the acoustic model (Zen et al., 2013; Ren et al., 2020). Additional prosody models are becoming common in the back-end to predict a prosodic specification for the acoustic model (Wan et al., 2019; Shechtman and Sorin, 2019).

For SPSS, the front-end is mostly used to predict discrete aspects of prosody, such as pitch accent and pause placement (Rosenberg, 2010; Rendel et al., 2017; Klimkov et al., 2017), whereas the back-end typically predicts more fine-grained and continuous representations of prosody such as $F_0$ (Wang, 2018; Ronanki, 2019), intensity (Ren et al., 2020; Mohan et al., 2021), and durations (Chen et al., 2017; Henter et al., 2017b). For SPSS, the front-end also extracts other, prosodically relevant, features such as: syntactic structure, syllabification, and word-class or part of speech (Black et al., 1998). These can be used by the back-end as context information when predicting prosody.

This is a fragmented approach to prosody modelling; some aspects are predicted in the front-end and others in the back-end. This will lead to inefficiencies as errors are propagated, and information provided to, or learnt by, one model is not necessarily shared. Prediction or extraction of features in the front-end of-

ten uses simple heuristic rules or simple models. Most notably, pause placement is often determined by punctuation (Black et al., 1998), this greatly limits the acoustic model's use of prosodic phrasing.

S2S models have moved away from this fragmented approach by relying on a more minimal front-end, using phone identity as the only input (Shen et al., 2018). Any additional features are extracted or predicted in the back-end. Prosody has been predicted explicitly in S2S using a separate prosody model (Wan et al., 2019; Shechtman and Sorin, 2019) and jointly as part of the acoustic model (Ren et al., 2020). However, these models don't necessarily improve on the existing limitations of prosody models; we must address the limitations explicitly. For example, we can mitigate the lack of context information by providing additional information: Tyagi et al. (2020) models surrounding utterances together in the front-end; while my work in Chapter 6 introduces additional context features to the prosody model.

Separate prosody models are most often used to predict $F_0$, durations, and pause placement. As early as the 1980s, $F_0$ was modelled with neural networks (Scordilis and Gowdy, 1989; Traber, 1990), while early work on phrase break prediction used classification and regression trees (CART) (Ostendorf and Veilleux, 1994; Hirschberg and Prieto, 1996). In HMM-based SPSS, durations were modelled with phone-dependent Gaussian or gamma distributions (Zen et al., 2007), and $F_0$ was modelled as a component of the HMM emission distribution (Zou et al., 2010; Qian et al., 2010).

In NN-based SPSS, neural networks have been used widely for $F_0$ modelling (Fernandez et al., 2014; Wang, 2018). Latorre and Akamine (2008) explicitly model $F_0$ variation hierarchically over multiple prosodic domains. Feature streams can also be modelled hierarchically, first predicting the voiced-unvoiced decision followed by $F_0$ in voiced regions (Lei et al., 2010; Wang et al., 2017a). Alternatively, $F_0$ can be interpolated during unvoiced regions and modelled as a continuous signal (Yu and Young, 2011). Some methods learn discrete representations of $F_0$, inspired by the discrete structure intonation exhibits (Ronanki et al., 2016a; Wang et al., 2019b). Recent S2S models have explicitly predicted $F_0$ either as a separate task (Arik et al., 2017b,a) or as part of a single, jointly trained model (Ren et al., 2020).

Duration models in NN-based SPSS use NNs (Zen et al., 2013). These models assume a Gaussian structure, yet phone durations are non-Gaussian—segment durations form a right-tailed gamma distribution. Henter et al. (2016) and Ronanki et al. (2016b) solved this using a non-parametric approach, while Chen et al. (2017) use a discrete random variable to model durations. In S2S models, duration is often modelled implicitly with attention. However, many approaches take advantage of the monotonic alignment between phones and speech using attention variants (Lim et al., 2020; Shen et al., 2020) or an explicit duration model (Yu et al., 2019; Ren et al., 2020).

In phrase break modelling, much of the difficulty lies in defining phrase break labels for supervision, since annotation is expensive. Often, pauses above a threshold are used to capture more salient phrase breaks (Rendel et al., 2017; Klimkov et al., 2017). Mishra et al. (2015) explored using additional context specifically for phrase break prediction, including part of speech and dependency relations. Dall et al. (2014) investigate the importance of filled pauses and demonstrate improved pause insertion using an RNN and an n-gram language model.

Prosody modelling can also be improved using new feature representations, such as wavelets. Wavelets are a generic representation of temporal signals (Mallat, 1989). The wavelet transform is function decomposition that produces a "scalogram", this is analogous to how the Fourier transform produces a spectrogram. While a spectrogram represents a signal in frequency space, the scalogram represents a signal by the contribution of different scales of a template signal: the mother wavelet. Decomposing by scales makes the wavelet transform useful for prosodic signals as prosody operates over different domains (Vainio et al., 2013; Ribeiro and Clark, 2015). Suni et al. (2017) proposed an approach to represent intonation, loudness, and duration within a single scalogram—this would allow for prosody control over multiple domains with a single representation.

We have covered a broad background on both TTS and prosody. However, three important topics remain. In the following section, I discuss evaluation of TTS and prosody. In Section 2.4, I discuss important considerations regarding speech data. Finally, I introduce the machine learning methods used in this thesis, in Section 2.5.

## 2.3 Evaluation

Evaluation is important for any machine learning system. For generation tasks, including speech synthesis and prosody synthesis, the quality of an output is subjective, i.e. the quality depends on an individual's opinions.

There are two broad approaches to speech evaluation: objective and subjective. **Objective evaluations** refer to metrics that can be computed algorithmically, in contrast with **subjective evaluations** which rely on human judgements. In TTS, it is common to evaluate intelligibility and naturalness. Other aspects, such as speaker similarity (Wester et al., 2016), likeability (Campbell, 2007, Section 3.2), and cognitive load (Govender and King, 2018) can be important, however they are less relevant to this thesis. To evaluate prosody quality, we must consider how appropriate a rendition is for a given context.

**Intelligibility**—how accurately listeners understand the lexical content—is solved for high-quality data in high-resource languages for non-technical language when listeners have normal hearing and are listening in quiet conditions. I do not conduct any intelligibility evaluations, though evaluating intelligibility is still important for certain data, languages, or listeners, and during initial testing of a voice.

Naturalness is often considered as an overall measure of how close synthetic speech is to human speech, covering both acoustic and prosodic quality. Yet, as explored in Chapter 3, state-of-the-art synthetic voices can be highly natural while having unsatisfactory prosody. This is possible as the pronunciation, articulation, and acoustic fidelity are all satisfactory, but the prosody is flat and not chosen specifically for any context.

Naturalness and appropriateness are hard to separate in evaluations.[3] To provide a clear distinction between these two related concepts, I define **naturalness** as the similarity of acoustic quality to human speech, which includes acoustic artefacts, background noise, audio fidelity, as well as the lack of unrealistic prosody (cf. Chapter 3). **Appropriateness** measures how suitable the prosody is for a given context and is discussed in Section 2.3.3.[4]

[3]Recent work explicitly demonstrated that naturalness and appropriateness are fundamentally different (O'Mahony et al., 2021).

[4]While definitions in the literature may not make the distinction between naturalness and appropriateness, it is overwhelmingly common to evaluate single out-of-context sentences. Such

Naturalness is still an open challenge, though recent work, using S2S models (Ren et al., 2020) and neural vocoders (Kalchbrenner et al., 2018), has greatly improved acoustic quality.

For mean-opinion score tests, the naturalness of state-of-the-art systems is now high enough that the difference with human speech can't be precisely measured (Jia et al., 2021; Elias et al., 2021). This is not to say that TTS has reached parity with human speech, but that certain testing paradigms are no longer sufficient. Naturalness is still an important aspect of TTS performance, and more precise evaluations should be used. However, more attention is being given to the evaluation of appropriateness in TTS (Latorre et al., 2014; Mendelson and Aylett, 2017; Loupi, 2017; Clark et al., 2019; Wagner et al., 2019). As naturalness reaches parity with human speech, the next task for achieving general-purpose TTS is prosody synthesis. Unfortunately, evaluation of prosody is not straightforward (Clark et al., 2019).

### 2.3.1 Objective evaluation

Objective evaluations are reasonably accurate in measuring some aspects of TTS quality, such as pronunciation quality or pronunciation errors (Janssen, 1957; Steeneken and Houtgast, 1980; Taghia and Martin, 2013). For high-resource languages, intelligibility can be reliably evaluated with ASR (Karbasi and Kolossa, 2017). However, for many aspects of speech, objective metrics are not guaranteed to correlate with human perception—hence the need for subjective evaluations. In SPSS, objective metrics that evaluate predicted vocoder features are useful as a debugging tool, these are introduced in Chapter 4.

There do exist approaches to replicate perceptual measures of naturalness using objective methods. The most notable is Perceptual Evaluation of Speech Quality (PESQ) (Recommendation P.862, 2001). PESQ was designed for telephone speech, and while Cernak and Rusko (2005) did demonstrate correlation of PESQ with subjective results for synthetic speech, it is too weak to make PESQ a reliable replacement for subjective tests.

Recent work has taken a more direct approach to replicating subjective re-

evaluations cannot determine if the prosody is appropriate, but only if it is invalid. Therefore, evaluations with single out-of-context sentences will mainly capture naturalness as defined here: acoustic quality and prosodic "plausibility".

sults; Patton et al. (2016) and Lo et al. (2019) train neural networks to predict naturalness results from synthetic speech using human ratings as targets. These models are very noisy on a per-utterance or per-listener basis, but predictions are usable at the system level. Such models must be used within domain, i.e. using the same dataset and performing the same task (e.g. voice conversion). Ultimately, objective evaluation is not yet able to replace the quality of feedback human participants provide, especially for evaluation of naturalness and appropriateness.

### 2.3.2 Subjective evaluation

Subjective evaluations attempt to measure the opinion of users, often to assess quality or preference. While evaluation tasks can have objectively correct answers, such as a transcription task, human responses depend on many factors and will always be subjective. The end-users for TTS voices are humans, meaning there is no better proxy than asking listeners directly. However, human perception is subjective, meaning judgements in listening tests will be noisy (Roy et al., 2017), thus it is important that listening tests are designed carefully and that statistical significance is tested (Wester et al., 2015). By using many listeners, a subjective evaluation aims to capture the wisdom of the crowd.

The statistical tests used for evaluations depends on the task. Where relevant, I introduce the statistical tests performed within each chapter. If many statistical hypotheses are being tested then multi-test correction must be used (Wester et al., 2015). For a given p-value, say 5%, 1 in 20 significant statistical inferences within a single study will be erroneous inferences. By using multi-test-correction, this ensures that 1 in 20 studies will contain an incorrect statistical inference, as opposed to 1 in 20 incorrect inferences within a single study (Holm, 1979).

Depending on what is being evaluated, different variables must be controlled. In naturalness tests, linguistic content is typically a control variable. This can be enforced by using a variety of sentences and having each participant hear every sentence. If this would make the listening test too long, a Latin square design can be used (Kirk, 2013, Chapter 14). By treating a group of participants as a single "virtual" participant, a Latin square design can control for the necessary variables for each virtual participant.

A major design choice for listening tests is the task: the structure in which listeners can respond. This may simply be a Likert scale (Likert, 1932) as used in mean-opinion score (MOS) tests (Recommendation P.85, 1994). MOS tests present a single stimulus at a time and ask for a rating independently of other stimuli. For state-of-the-art S2S voices, MOS tests lack the precision (i.e. listener agreement) to demonstrate differences in system performance. This can be observed in the results of many papers where confidence intervals commonly overlap with natural speech (Shen et al., 2018, 2020; Jia et al., 2021; Elias et al., 2021). This is not to say that these voices are equal in quality to human speech, but that these studies should use tasks that produce more precise results.

The preference test, or "AB" test, directly compares stimuli from two systems. Due to this, preference tests have a higher precision than MOS tests, which rate stimuli separately. A single preference test is restricted to comparing exactly two systems, however, there are many situations where more than two systems need to be compared on a single axis. Fortunately, by performing multiple preference tests—one for each pair of systems—it is possible to combine the results and provide a rating for each system on a single axis. Multiple preference test results can be combined using multi-dimensional scaling (Borg and Groenen, 2003) or ordinary least squares (proposed in Chapter 3). The cost of this approach is quadratic in the number of systems as it requires a preference test for each pair of systems, i.e. systems must be compared combinatorially. Despite the quadratically increasing cost of this approach, preference tests are the only task that can scale to many systems without sacrificing precision or accuracy. This approach does not sacrifice precision as stimuli are always compared directly. Accuracy is not sacrificed as the task is comparative, unlike MOS tests, and is simpler for listeners than a MUSHRA test with many systems.

The multiple stimuli with hidden reference and anchor (MUSHRA) test (BS Series, 2014) is designed to directly compare multiple systems, unlike the preference test. It was developed for evaluation of speech coding algorithms. MUSHRA is comparative, like preference tests. Additionally, participants must identify one stimulus as the anchor (i.e. a lower-bound) and one stimulus that matches a visible reference (i.e. an upper-bound). This ensures the rating scale is normalised per listener, instead of solely relying on labels. MUSHRA tests typically use a scale between 0 and 100. This allows participants to represent relative differ-

ences between stimuli. Unlike preferences tests which require a quadratic number of pairwise tests to compare multiple systems, MUSHRA relies on participants performing a quadratic number of comparisons when rating. For this reason, increasing the number of systems above 4 or 5 makes the task more difficult and will impact the test's precision and accuracy.

In TTS, variants of MUSHRA are relatively common, including: removing the requirement to identify the lower-bound or upper-bound, removing the visible reference, or using an ordinal scale to directly capture rankings instead of ratings. By deviating from the original evaluation spec (BS Series, 2014), these modifications may impact the reliability of the test. However, to explore a given hypothesis, it can be more important to modify the test. Ideally, substantial changes should be validated to ensure the test remains reliable.

Other listening tests with different tasks include: forced-choice annotation (e.g. labelling tasks to evaluate interpretability), free-form annotation (e.g. transcription to evaluate intelligibility), or free-form interviews for phenomena that are harder to codify. Changing the structure in which participants respond also impacts how listeners behave.

An evaluation's question or prompt also has a major effect on its efficacy. When using keywords, such as intelligibility, naturalness, or likeability, it is important to consider the varying interpretations participants might hold. While complicated technical terms can be explained, participants' interpretations are more likely to vary if lengthy explanations are required. Other detail can also be presented to influence what information the test is capturing, such as: the purpose of the evaluation, surrounding sentences and audio, or other modalities like text and video.

### 2.3.3 Prosody evaluation

Evaluating synthetic prosody is challenging as a sentence can have multiple renditions, only some of which will be suitable for the context (Latorre et al., 2014). A prosodic rendition can only be considered "good" if it is suitable for the given context. This measure of prosody quality is **appropriateness**. Without the context we cannot know if a prosodic rendition is appropriate or not. In addition, many TTS systems use isolated utterances and do not provide control of prosody.

Such TTS systems cannot change an utterance's prosodic delivery and are unable to intentionally produce appropriate prosody.

While the context needed to predict prosody is very broad, subjective evaluations require less context. In subjective evaluations of prosody, we can rely on the human participants to infer missing contextual information given some limited context. To evaluate the appropriateness of an utterance, the context can be explicit, such as a description of a situation (Goodhue et al., 2016), or implicit, such as the surrounding sentences (Clark et al., 2019).

Explicit context that describes a situation must be carefully designed, and conceiving of sentence-context pairs is time consuming. In addition, the written description of the context must be concise, so that participants can quickly and correctly understand the situation. Goodhue et al. (2016) study which prosodic constructions are appropriate for different explicit contexts. For the phrase "You like John." they describe three contexts: when the interlocutor failed to mention they like John, when the interlocutor falsely claimed they didn't like John, and when the interlocutor falsely claimed they did like John. They demonstrate that multiple prosodic constructions can be appropriate for each context, and that the most commonly used construction is different for each context.

While highly specific sentence-context pairs like those from Goodhue et al. (2016) are not used in TTS evaluations, much simpler situations are used to evaluate controllable TTS systems. For emotion control, asking "Does the stimulus sound happy?" provides explicit context, i.e. this situation calls for a happy response. Clearly this context is limited, but it enables evaluation of the prosodic variation the system is designed to control.

A simpler approach to defining context is to provide participants with surrounding context. In this way, the context is implicit and must be interpreted by participants. However, evaluating long sequences of utterances is not without challenges (Clark et al., 2019). For example, cognitive biases can impact evaluations. The recency bias, where recent events are more salient, and the primacy effect, where initial events are more salient, can both impact the judgements provided by listeners for long stimuli (Deese and Kaufman, 1957). While the anchoring effect can make it difficult to present multiple renditions to listeners—anchoring refers to the bias where judgements are influenced by a, possibly arbi-

trary, reference point (Sherif et al., 1958). Unfortunately, there is little research on the efficacy of prosody evaluations that use longer stimuli or provide surrounding context.

A complementary school of thought says evaluations should be designed according to the intended use-case (Campbell, 2007; Mendelson and Aylett, 2017; Wagner et al., 2019). Campbell (2007) argues that we need to evaluate how end users experience and perceive our systems, not how well our technology improves upon a previous approach. Mendelson and Aylett (2017) focus on the importance of the end use-case and propose an interactive evaluation paradigm to conduct conversations using a Wizard-of-Oz design. This places the whole system within the context of its intended application which should lead to more relevant findings. This interactive approach to evaluation can be seen as an extension of explicit and implicit context, where the explicit context is the goal of conducting a conversation, and the turns of the conversation are the implicit context.

It is important to consider listener agreement in any evaluation. Human perception is subjective, and listeners can form different interpretations of the task instructions. This can be observed for ToBI annotation: despite the detailed instructions, training, and examples, ToBI annotations have low inter-annotator agreement (Syrdal and McGory, 2000). Human perception of prosody can be affected by what listeners are told to attend to, the context made available, and the type of content. Cole et al. (2014) found that listeners varied their prominence judgements depending on whether they were told to attend to an utterance's meaning or just its acoustics. Turnbull et al. (2017) found that listeners were more likely to identify certain accented words as prominent in contrastive contexts, embedded in dialogues. While Hinterleitner et al. (2011) find that content type—e.g. long sentences, direct speech, poetic, action, or children's books—has a significant effect on the perception of: speech pauses, intonation, emotion, and stress. Additionally, different listener demographics can exhibit different behaviour. Klimkov et al. (2017) found that vetted listeners (i.e. employees) were consistently less decisive than listeners from Amazon Mechanical Turk, using the "no preference" option more frequently. Many factors can impact human behaviour, but ultimately judgements from different listeners are inherently noisy (Roy et al., 2017).

While evaluating "good" prosody, i.e. appropriateness, is challenging, it is

much more straightforward to evaluate "bad" prosody. Obvious prosodic mistakes can be evaluated without the need for context by using naturalness as a proxy. Such an evaluation will indicate unrealistic prosody, but will also capture participants' perception of other factors, such as pronunciation quality, acoustic quality, and likeability (Campbell, 2007). As discussed in Chapter 3, unrealistic prosody corresponds to the prosody humans can't, won't, or shouldn't produce in any context. Unrealistic prosody evaluation can use isolated utterances, making it more straightforward than appropriateness evaluations. Investigating when and where systems make mistakes can be very informative for making improvements. However, this is not a replacement for measuring appropriateness.

The focus of this thesis is not prosody evaluation. However, in investigating the absence or use of context in prosody synthesis, it is of course necessary to evaluate prosody. Since prosody evaluation is still an unsolved problem, in each chapter I design novel listening tests to answer specific questions. I also rely on existing approaches, such as naturalness tests, where applicable.

## 2.4 Data

Data is the backbone of modern TTS systems as they rely on neural networks. Model improvements can be made to improve performance, robustness, and data efficiency. However, neural networks may be unable to produce a behaviour unless it is exhibited by the data (Jacot et al., 2018; Domingos, 2020). For TTS, there is increasing evidence that new styles can be produced if data in that style is available (Prateek et al., 2019; Cotescu et al., 2019). Thus, the style and content of a TTS dataset is especially important.

### 2.4.1 Quality and variability

There are many desirable and undesirable traits of speech data. The status quo for commercial synthetic voices is to use "clean" data—having no background noise and being recorded with high-quality equipment. However, such data is typically elicited or acted, with limited range of styles and variability. Podsiadło and Ungureanu (2018) demonstrated, that for assistant-style speech data, state-of-the-art performance can be achieved using only ~10% of the data compared to what is used in previous systems. This suggests that there is little additional variation in the remaining ~90% of the data.

From a machine learning perspective, data with less variability is easier to model (Geman et al., 1992). However, the prosody in such data will not reflect natural human prosody. This trade-off between quality and variability is especially noticeable for long-form speech (Clark et al., 2019). For example, one early commercial smart home assistant used assistant-style TTS voices to read out a morning news briefing, impressionistically this resulted in an uninteresting user experience. The feature's synthetic speech was later replaced with human-read briefings.[5] One solution to this poor listener experience is to collect data by eliciting the desired news-reading style (Prateek et al., 2019). While this approach resolves the issue, it is not a scalable solution.

### 2.4.2 Found data

Designing and collecting data that exhibits interesting prosodic behaviour is challenging (Goodhue et al., 2016). Typically, it requires the use of actors and careful script design to elicit certain behaviour (Douglas-Cowie et al., 2003). Instead, data can be sought out from other domains and re-purposed for TTS. Such data is known as found data. Many datasets used in academic TTS research are derived from found data, most notably: audiobooks (Stan et al., 2013; King et al., 2013; King and Karaiskos, 2016; Ribeiro, 2018a; Zen et al., 2019) and podcasts (Lotfian and Busso, 2017; Székely et al., 2019; Clifton et al., 2020).

Found data often contains much more realistic speech. However, this may be at the expense of acoustic quality. Found monologue or dialogue data can have various challenging qualities, including: background noise (Canavan and Zipperlen, 1997; Ito, 2017; Hernandez et al., 2018; Zen et al., 2019), distant speakers (Carletta, 2006), overlapping speakers (Canavan and Zipperlen, 1997; Carletta, 2006), and disfluencies (Carletta, 2006; Székely et al., 2019). While audiobooks are a high quality source of found data, there may also be channel variability across different speakers or recordings; this is easily avoided when collecting new data.

Prosody in audiobooks is distinct from prosody in dialogues, as the communicative task is different. While both are challenging, the importance of taking

[5] No references or news articles could be found. The failure was caused by the intrinsic style of the data, as evidenced by Podsiadło and Ungureanu (2018) and Prateek et al. (2019). Modelling assumptions that lead to average prosody may also contribute.

turns and interpersonal relations, among other things, means that dialogue data presents an even more challenging prosody synthesis task for TTS.

One approach to dealing with the variable quality of found data is to filter out problematic data (Baljekar and Black, 2016; Gallegos et al., 2020). Information such as low signal-to-noise ratio might indicate bad data, but could be caused by overlapping speech which can be related to back-channelling. Filtering out data that contains back-channelling means removing realistic and interesting prosody. Ideally, any filtering should avoid removing interesting data.

### 2.4.3 Usborne children's audiobook dataset

In Chapters 3, 4, and 5, I use the Usborne children's audiobook dataset. This dataset was introduced for the Blizzard Challenge 2016 (King and Karaiskos, 2016), and was provided by Usborne Publishing. It consists of professionally-recorded audiobooks intended for a 4–6 year-old audience. The dataset is single speaker—a female speaker of standard southern British English—and contains 6.5 hours of speech, or roughly 7,250 sentences.

The choice of this dataset is motivated by the need for interesting and challenging prosodic variation. The data exhibits different modes of variation that are captured by the control mechanisms introduced in the following chapters. Stories in the dataset include: traditional stories (e.g. *Little Red Riding Hood*), simplified Shakespeare (e.g. *Macbeth*), and non-fiction (e.g. *The Story of Chocolate*). These are read in an expressive style, with various character voices and a substantial quantity of direct speech.

Additional datasets are used in Chapters 4 and 5, and a different TTS dataset is used in Chapter 6. These are introduced and discussed within the relevant chapters.

## 2.5 Machine learning

TTS relies on a number of machine learning techniques. Within the front-end, decision trees, linear regression, and logistic regression are all used (Black et al., 1998). In hybrid unit-selection and statistical parametric speech synthesis, hidden Markov models and decision trees were widely used. However, over the past

decade, neural networks have become the predominant technique for TTS (Watts et al., 2016, 2019). My research focuses on the back-end, specifically the acoustic and prosodic models. In the following sections, I introduce the machine learning techniques used in this thesis, which includes neural networks and graphical models.

### 2.5.1 Neural networks

Neural networks (NNs) have been studied in niche applications for decades (Rosenblatt, 1958; Rumelhart et al., 1986; Bishop et al., 1995). Following popularisation in computer vision, NNs have become pervasive (Krizhevsky et al., 2012), thanks to: increasing computational power, improvement of algorithms for efficient training, and the ability to train on successively larger datasets (Goodfellow et al., 2016). NNs are often seen as universal function approximators, able to learn any process given sufficient capacity and data (Csáji et al., 2001). Some theoretical research suggests that NNs may not be able to generalise past the provided data (Jacot et al., 2018; Domingos, 2020). This is relevant to prosody and style control as we must ensure the data contains the variation we are interested in synthesising (Podsiadło and Ungureanu, 2018; Prateek et al., 2019).

In their simplest form, NNs are very similar to logistic regression: a learnt projection followed by a non-linearity. Unlike logistic regression, where a projection matrix, or weight matrix, that minimises the loss can be analytically computed, NNs use optimisation algorithms to learn the weight matrices. Typically, optimisation uses stochastic gradient descent (SGD), or a similar variant, where the weights are iteratively updated in small steps to improve performance on a subset of the data—a "mini-batch". Compared to logistic regression, NNs can use a variety of non-linearities, not just the sigmoid function.

Designing a NN involves two main components: architecture and loss. Architecture includes the choice of layers and activation functions. Choosing or designing a loss is very important, though in many applications there may be one obvious choice, such as cross-entropy for classification. However, designing a loss determines how the model behaves and what it learns. The loss design is especially important for self-supervised learning.

Other aspects, such as the optimisation routine, may be considered integral

to designing a model. However, off-the-shelf algorithms are relatively robust, such as adaptive variants of SGD (Kingma and Ba, 2014), random initialisation (Glorot and Bengio, 2010), and backpropagation (Rumelhart et al., 1986). Models are still sensitive to the learning rate, and empirically choosing the learning rate or learning rate schedule is often very important to successful training.

The data also has an effect on the resulting model—arguably it has the greatest effect. Choosing the data, as well as cleaning and filtering it is important. Fortunately, the datasets used in this thesis are high quality, and have been used extensively in the literature, thus data cleaning and filtering are not considered here.

A model's architecture is typically a sequence of layers and activation functions applied consecutively. However, the forward pass in a model can be more complex, such as for S2S models where auto-regression or upsampling are necessary. The forward pass for most layers can be parallel (sometimes known as "feed-forward") or auto-regressive depending on the application, although recurrent models by design must be autoregressive. Autoregressive models using recurrent, convolutional, or attention layers have all demonstrated state-of-the-art performance for modern TTS (Wang et al., 2017b; Tachibana et al., 2018; Li et al., 2019).

The model architecture and loss can impose various inductive biases. An **inductive bias** is any assumption made by the model that allows it to generalise beyond the training data (Mitchell, 1980). This can be as simple as restricting a classification model to a fixed set of target labels. The main layer types—dense, recurrent, convolutional, and attention-based—all impose structural inductive biases. Especially for self-supervised models, the loss plays an important part in defining what the model learns. For example, the masked language modelling and next sentence prediction losses used in BERT (Devlin et al., 2019) impose inductive biases towards understanding semantics, syntax, and context, both short and long range.

#### 2.5.1.1 Dense layers

The first incarnation of neural networks—originally a computational analogue for neurons in the brain—used dense layers (Rumelhart et al., 1986). Dense layers

consider how each input contributes to each output through an affine transform. Dense layers impose no structural assumptions, this makes learning less efficient for complex data like speech. However, deep networks with many layers impose an inductive bias for hierarchical and distributed representations (Bommasani et al., 2021, Section 4.1).

#### 2.5.1.2 Recurrent layers

For sequence data, it is important to take into account the temporal relationship between time-steps. A recurrent layer combines information from the previous time-steps with the input at the current time-step (Rumelhart et al., 1985). The simplest recurrent layer is the repeated application of a dense layer across time, with feedback of each time-step's output to the next time-step's input. This iterative design allows recurrent layers to use variable length sequences. Recurrent neural networks (RNNs) are models that use at least one recurrent layer. RNNs impose a temporal inductive bias, making them better at learning temporal patterns.

More complex recurrent layers were introduced to resolve issues such as vanishing gradients, including: the long short-term memory (LSTM) cell (Hochreiter and Schmidhuber, 1997), and the gated recurrent unit (GRU) (Cho et al., 2014). LSTMs and GRUs use gates (i.e. the sigmoid function) to control what information is used or discarded. These gates allow the model to use the input to adapt how it processes that same input. This imposes an inductive bias to dynamically adapt model behaviour to the current input. Variants such as bi-directional RNNs allow recurrent models to incorporate past and future information (Schuster and Paliwal, 1997). Another variant, the clockwork RNN, can operate at different time scales (Koutnik et al., 2014). Clockwork RNNs allow for hierarchical prosody modelling over linguistic domains (Wan et al., 2019).

#### 2.5.1.3 Convolutional layers

Convolutional neural networks (CNNs) use the mathematical convolution operation to extract local information from the input (Lecun et al., 1998).[6] Convolutions use a kernel to extract this information. Importantly, this kernel is

[6] While neural network convolutions actually correspond to mathematical cross-correlation, the difference is unimportant because the kernel is learnt.

reused across all points across the input: it is "convolved" over the input. Unlike human-defined kernels from traditional computer vision (Sobel and Feldman, 1968), CNNs use stochastic gradient descent to learn the kernel. When many convolution layers are stacked, early layers have been shown to learn low-level information, while later layers learn more abstract high-level information (Erhan et al., 2009). CNNs impose a structural inductive bias: they assume that local regions of input contain the most relevant information, this can be in terms of spatial, temporal, or graph distance. This is a more efficient parameterisation for data with local structure. In addition, convolutions can be computed in parallel, unlike recurrent layers, making their time complexity lower.

Convolutions are relatively limited in their receptive field: the size of the input accessible from a given position in the output. However, van den Oord et al. (2016) solved this problem by applying multiple dilated convolutions (Holschneider et al., 1990) allowing the receptive field to scale exponentially with the number of layers. Other convolution variants can be used to make training and inference more efficient, such as depth-wise separable convolutions (Sifre, 2014; Kang et al., 2021).

#### 2.5.1.4 Attention

Attention was designed to handle sequence data for tasks where the input and target have different sequence lengths (Bahdanau et al., 2014). Attention learns to compute scores that represent the relevance of different input time-steps to each output time-step. For example, in machine translation we want to know which source words help determine each translated word. However, by considering all input-output pairs there are $\mathcal{O}(N^2)$ scores to compute, making attention computationally expensive. Attention was initially only applied to align two sequences of a different length, self-attention is the application of attention when the input and output have the same sequence lengths (Vaswani et al., 2017). Vaswani et al. (2017) also introduced multi-headed attention where multiple context vectors are computed, each based on separate scores.

Attention is able to share information from arbitrarily far away from the current output time-step using the context vector, unlike RNNs where information must be propagated step-by-step through the recurrent state. This resolves issues with vanishing gradients in time. Compared to RNNs, attention imposes an

inductive bias on the importance of long-range dependencies.

Attention scores are computed dynamically based on the inputs. This provides an inductive bias for dynamic processing of inputs, similar to the dynamic gating in LSTMs, but more powerful. It is also conceptually similar to meta-learning, where a model is trained to predict network parameters for another model (Vilalta and Drissi, 2002). In a sense, attention is a convolution with arbitrary width and a meta-learnt kernel.

#### 2.5.1.5 Regularisation

Neural networks are a powerful machine learning technique thanks to their flexible parameterisation. However, due to their large representational capacity, they are prone to relying on the training data too much. In the worst case, the model will memorise data. If the model "overfits" to the training data in this way, it will not generalise well to unseen data. Fortunately, a common cause of this behaviour is very large weights, which can be regularised by penalising large weights. The most common form of regularisation is to add an additional term to the loss, typically the $L2$ norm of the weights (Ng, 2004). Minimising this jointly with the loss will minimise the size of the weights, and thus mitigate some forms of overfitting.

However, to resolve overfitting more generally (i.e. the use of misleading patterns) other forms of regularisation are required, such as dropout, data augmentation, and multi-task learning (Goodfellow et al., 2016, Chapter 7). Dropout randomly removes parts of a layer's input, this adds noise to the training data making overfitting less likely (Srivastava et al., 2014). Data augmentation is commonly used in computer vision, where it improves position and rotation invariance (Taylor and Nitschke, 2018). Multi-task learning adds additional regression or classification tasks to a model (Caruana, 1998). The correct additional task can focus the model on extracting desirable information from the inputs, thus reducing the chance of overfitting on misleading information.

#### 2.5.1.6 Data normalisation

It is common practice in TTS, and in most fields, to normalise data streams according to global statistics. This is important as neural networks train more efficiently on normalised data (Bishop et al., 1995, Chapter 8). The type of

normalisation to perform depends on the data. For real valued data, mean-variance normalisation is common. While for ordinal integer data, such as linguistic counter features (cf. Table A.1, pp. 168), min-max normalisation can be used.

### 2.5.2 Graphical models

Graphical models, also known as probabilistic models, provide a powerful framework for capturing latent information and incorporating distributional and structural assumptions. While neural networks can impose inductive biases through various design choices, it is difficult to encode prior knowledge or assumptions about the underlying process that generated the data. For example, in spectrograms of speech we have prior knowledge that the higher frequencies will contain more noise than the lower frequencies, this type of knowledge can be encoded in a graphical model.

Graphical models aim to learn the true distribution of the data, $p^*(x)$. By using this formalism, it is possible to impose independence assumptions, e.g. in unvoiced regions we could assume the spectrogram is independent of $F_0$. The conditional independence and dependence relationships derived from these assumptions are represented as a graph, hence the name graphical models. Common representations for graphical models include: belief networks, Markov networks, factor graphs, and junction trees (Barber, 2012).

In addition to modelling the data distribution, graphical models can model latent (i.e. unobserved) factors that correlate with observed behaviours. This allows graphical models to explicitly capture unexplained variation, such as prosody (cf. Chapters 3 and 5). There are a broad variety of graphical models that model latent variables. The hidden Markov model, used for SPSS, is a classic example of a latent variable model. Recent research has resulted in more powerful graphical models, such as variational autoencoders (Kingma and Welling, 2013), normalising flows (Dinh et al., 2017; Papamakarios et al., 2021), and diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020). These models are powerful in part thanks to their complex parameterisation: large neural networks are used to predict the parameters of the distributions that define these graphical models.

Learning the parameters of a graphical model's distributions can be simple,

such as for naïve Bayes, where maximum likelihood estimation corresponds to counting occurrences in the data. While for other graphical models, especially latent variable models, exact inference may not be possible. Learning algorithms such as expectation maximisation, or its more general derivation, variational inference, can be applied to learn an approximation of the true model.

Graphical models use data to learn their distributions' parameters. However, we only observe a finite number of data points, i.e. samples from the true data distribution, $p^*(x)$. Fundamentally, this means modelling $p^*(x)$ is intractable. **Variational inference** avoids the need to directly model $p^*(x)$ by introducing a variational distribution $q(x)$ (Jordan et al., 1999). Instead of modelling $p^*(x)$ or minimising the divergence between $q(x)$ and $p^*(x)$ directly, we can derive a variational lower bound on $p^*(x)$. Maximising this lower bound brings the variational approximation, $q(x)$, closer to the true data distribution, $p^*(x)$.

#### 2.5.2.1 Variational autoencoders

This thesis uses variational autoencoders, a graphical model that learns a latent distribution using variational inference (Kingma and Welling, 2013). **Variational autoencoders** (VAEs) attempt to learn an approximate posterior $q_\phi(z \mid x)$, where $z$ is the latent random variable being learnt (Doersch, 2016). This latent variable represents the information required to reconstruct the data. Our aim is to minimise the dissimilarity between the approximate posterior (i.e. the variational distribution) $q_\phi(z \mid x)$, and the true posterior $p_\theta(z \mid x)$. Therefore, to derive the variational lower bound for VAEs, we start with the following divergence definition,

$$D_{KL}(q_\phi(z \mid x) \,||\, p_\theta(z \mid x)) = \mathbb{E}_{\tilde{z} \sim q_\phi(z|x)} \left[\log q_\phi(\tilde{z} \mid x) - \log p_\theta(\tilde{z} \mid x)\right] \tag{2.1}$$

However, we do not know the true posterior $p_\theta(z \mid x)$; this is what we are trying to approximate using variational inference. Therefore, we deconstruct the KL divergence using Bayes rule, resulting in three terms in Equation 2.5: a KL divergence between the approximate posterior and a prior term, $p(z)$; a

"reconstruction" term; and the data log likelihood.

$$D_{KL}(q_\phi(z \mid x) \,||\, p_\theta(z \mid x)) = \mathbb{E}_{\tilde{z}\sim q_\phi(z|x)}\left[\log q_\phi(\tilde{z} \mid x) - \log\left(\frac{p_\theta(x \mid \tilde{z})p_\theta(\tilde{z})}{p_\theta(x)}\right)\right] \tag{2.2}$$

$$= \mathbb{E}_{\tilde{z}\sim q_\phi(z|x)}[\log q_\phi(\tilde{z} \mid x) - \log p_\theta(x \mid \tilde{z}) - \log p_\theta(\tilde{z}) + \log p_\theta(x)] \tag{2.3}$$

$$= \mathbb{E}_{\tilde{z}\sim q_\phi(z|x)}[\log q_\phi(\tilde{z} \mid x) - \log p_\theta(\tilde{z})] - E_{\tilde{z}\sim q_\phi(z|x)}[\log p_\theta(x \mid \tilde{z})] + \log p_\theta(x) \tag{2.4}$$

$$= D_{KL}(q_\phi(z \mid x) \,||\, p_\theta(z)) - E_{\tilde{z}\sim q_\phi(z|x)}[\log p_\theta(x \mid \tilde{z})] + \log p_\theta(x) \tag{2.5}$$

The divergence term allows us to enforce structure using the prior, and is discussed below, the reconstruction term represents an autoencoding task, and the log likelihood is what we're ultimately interested in modelling accurately. Rearranging, we get the evidence lower bound (ELBO) for the log likelihood on the right hand side,

$$\log p_\theta(x) - D_{KL}(q_\phi(z \mid x) \,||\, p_\theta(z \mid x)) = \mathbb{E}_{\tilde{z}\sim q_\phi(z|x)}[\log p_\theta(x \mid \tilde{z})] - D_{KL}(q_\phi(z \mid x) \,||\, p_\theta(z)) \tag{2.6}$$

$$\log p_\theta(x) \geq \mathbb{E}_{\tilde{z}\sim q_\phi(z|x)}[\log p_\theta(x \mid \tilde{z})] - D_{KL}(q_\phi(z \mid x) \,||\, p_\theta(z)) \tag{2.7}$$

This is a lower bound as $D_{KL}(q_\phi(z \mid x) \,||\, p_\theta(z \mid x))$ is strictly non-negative. However, two challenges remain: we need to compute the expectation over $z$ for both terms, and we need to differentiate through the sampling process required by the expectation.

The stochastic gradient variational Bayes (SGVB) estimator uses two tricks to resolve these issues—SGVB is the method used to train VAEs by gradient descent. First, Monte Carlo sampling, with $S$ samples, is used to approximate the ELBO. Kingma and Welling (2013) found empirically that, for large enough mini-batches, Monte Carlo sampling with one sample (Equation 2.9) could be used.

$$\mathbb{E}_{\tilde{z}\sim q_\phi(z|x)}[\log p_\theta(x \mid \tilde{z})] - D_{KL}(q_\phi(z \mid x) \,||\, p_\theta(z))$$
$$\approx \frac{1}{S}\sum_{s=1}^{S} \log p_\theta(x \mid \tilde{z}^{(s)}) - D_{KL}(q_\phi(\tilde{z}^{(s)} \mid x) \,||\, p_\theta(\tilde{z}^{(s)})) \quad \text{where } \tilde{z}^{(s)} \sim q_\phi(z \mid x) \tag{2.8}$$

$$\approx \log p_\theta(x \mid \tilde{z}) - D_{KL}(q_\phi(\tilde{z} \mid x) \,||\, p_\theta(\tilde{z})) \qquad \text{where } \tilde{z} \sim q_\phi(z \mid x) \tag{2.9}$$

Second, SGVB uses the reparameterisation trick to make Equation 2.9 differentiable, as sampling is non-differentiable. This involves using an auxiliary distribution, $\mathcal{N}(\epsilon; 0, 1)$, in the computational graph that is not on a path between the loss and parameters. Transforming samples from this auxiliary distribution is equivalent to sampling directly from $\mathcal{N}(\mu, \sigma^2)$. Importantly, this does not require us to differentiate through the sampling process.

$$\tilde{z} = \mu + \sigma\tilde{\epsilon} \quad \text{where } \tilde{\epsilon} \sim \mathcal{N}(0, 1) \tag{2.10}$$

Given these solutions, a VAE is trained by maximising the following approximation of the ELBO for observations $X = \{x^{(1)}, \ldots, x^{(N)}\}$,

$$\mathcal{L}(X; \phi, \theta) = \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(x^{(n)} \mid \tilde{z}) - D_{KL}(q_\phi(\tilde{z} \mid x^{(n)}) \mid\mid p_\theta(\tilde{z})) \tag{2.11}$$

where $\tilde{z}$ is sampled from $q_\phi(z \mid x)$ using $\mathcal{N}(\epsilon; 0, 1)$ and the reparameterisation trick. In practice, the distributions $q_\phi(z \mid x^{(n)})$ and $p_\theta(x^{(n)} \mid z)$ are parameterised using neural networks.

VAEs share a similar structure to autoencoders, using an encoder, $q_\phi(z \mid x^{(n)})$; a decoder, $p_\theta(x^{(n)} \mid z)$; and a reconstruction loss term, $\mathbb{E}_{\tilde{z} \sim q_\phi(z|x)}[\log p_\theta(x \mid \tilde{z})]$. Compared to autoencoders, VAEs explicitly model uncertainty in the latent space, and are built on the foundation of variational inference. However, autoencoders also have a probabilistic interpretation: the hidden representation is simply a distribution with unknown variance. This means we do not know if the variational bound is in fact being maximised in an autoencoder and we cannot enforce structure using a prior.

### The prior and posterior

Structuring the problem in this way makes optimisation tractable as we are computing the divergence with a prior, $p_\theta(z)$, as opposed to a divergence with the true posterior, as in Equation 2.1. The prior represents our assumptions about the structure of the true data distribution and it enforces this structure on the approximate posterior. The parametric form of both the prior and approximate posterior are design choices. It is common to use an isotropic Gaussian for both. However, it has been shown that a diagonal covariance matrix is too limiting for the approximate posterior (Dorta et al., 2018). The choice of prior depends on the application. If smoothness and a uni-modal peak is required, then a standard

normal, $\mathcal{N}(\mathbf{0}, \mathbf{I})$, can be used. As explored in Chapter 5, other traits, such as multi-modal structure, can be achieved through bespoke priors.

### Conditional VAEs

The latent variable will capture the most salient information for the task of reconstructing the data. The latent variable's dimension acts as an information bottleneck, putting a threshold on how salient information must be, to be captured by the latent variable. However, if only certain information is desired to be captured, conditioning can be used in a VAE to encourage disentanglement (Sohn et al., 2015). The conditional VAE, also referred to simply as a VAE, follows a similar derivation to the original VAE (Kingma and Welling, 2013). However, technically it is no longer an autoencoder architecture. The ELBO and likelihood for a conditional VAE are as follows,

$$\log p_\theta(x \mid c) \geq \mathbb{E}_{\tilde{z}\sim q_\phi(z|x,c)}[\log p_\theta(x \mid \tilde{z}, c)] - D_{KL}(q_\phi(z \mid x, c) \mid\mid p_\theta(z \mid c)) \quad (2.12)$$

$$\mathcal{L}(X, C; \phi, \theta) = \frac{1}{N}\sum_{n=1}^{N} \log p_\theta(x^{(n)} \mid \tilde{z}, c^{(n)}) - D_{KL}(q_\phi(\tilde{z} \mid x^{(n)}, c^{(n)}) \mid\mid p_\theta(\tilde{z} \mid c^{(n)})) \quad (2.13)$$

where $\tilde{z}$ is sampled from $q_\phi(z \mid x, c)$ using $\mathcal{N}(\epsilon; 0, 1)$ and the reparameterisation trick. This model will avoid incorporating information contained in $c$, into the latent variable, $z$. The training data consists of $X = \{x^{(1)}, \ldots, x^{(N)}\}$ and $C = \{c^{(1)}, \ldots, c^{(N)}\}$.

The original formulation of a conditional VAE conditions the encoder, $q_\phi(z \mid x, c)$; prior, $p_\theta(z \mid c)$; and decoder $p_\theta(x \mid z, c)$, on $c$. However, the encoder and prior do not need to depend on $c$. While removing either dependency will change the graphical model, it still has a valid probabilistic interpretation. However, the decoder must be conditioned on $c$, as this is after the information bottleneck. Without this, the conditioning will have no impact on disentanglement.

### Posterior collapse

VAEs have a common failure mode known as posterior collapse. This refers to the situation during training where the VAE's approximate posterior collapses onto the prior: $q_\phi(z \mid x) = p_\theta(z)$. If the approximate posterior is always exactly equal to the prior then it contains no information about the conditional variable $x$.

This degenerate case stems from the structure enforced on the approximate posterior. A VAE enforces structure on the variational distribution, aiming to make the aggregate posterior look as similar to the prior as possible. This is achieved through the KL divergence term in the ELBO (Equation 2.7). The optimisation performed by SGVB may find that reducing this term is more impactful to the overall loss than improving reconstruction performance.

A common solution to mitigate posterior collapse is to anneal the weight on the KL divergence loss term (Higgins et al., 2017). Down weighting the KL divergence at the start of training reduces the cost of encoding information in the latent space. This makes it possible for the model to learn a non-degenerate encoder. Later in training when the annealing finishes, it is less likely for posterior collapse to occur as the approximate posterior provides information useful for reconstruction.

This brings us onto the final section of this chapter. Having covered TTS, prosody, evaluation, data, and machine learning, I now wrap up by putting my work in context with the rapidly progressing field of TTS.

## 2.6 Research context

Speech synthesis research has moved very rapidly in the past 3-5 years, advancing from "statistical parametric" models to sequence-to-sequence (S2S) models (Watts et al., 2016, 2019) and from signal processing vocoders to neural vocoders (Zhou et al., 2020). In certain chapters, my work does not use state-of-the-art techniques. This is either due to the timing of when the research was conducted, or the focus on other aspects of experimental design. Here, I outline the relevant context of the techniques I use and their relation to the current research landscape.

I use statistical parametric speech synthesis (SPSS) in Chapters 3, 4, and 5. This research was performed between 2016 and 2019, while S2S models were popularised in 2017 (Wang et al., 2017b). As demonstrated by my research in Chapter 6 and other recent research (Yu et al., 2019; Ren et al., 2019; Lim et al., 2020; Ren et al., 2020), the attention in S2S can be replaced with duration prediction. This results in S2S TTS becoming structurally similar to SPSS. Therefore, it would be possible to apply the methods I propose for SPSS to S2S architectures.

I use signal processing vocoders in Chapters 3, 4, and 5. However, direct waveform generation (van den Oord et al., 2016) and neural vocoders (Shen et al., 2018) were popularised between 2016 and 2018. In order to evaluate prosody, listeners must be able to focus on this aspect of speech, i.e. without distractions from low signal quality or bad pronunciation. While neural vocoders will reduce the cognitive load incurred by signal processing artifacts, the prosodic variation observed in these chapters was salient to listeners, despite the use of traditional vocoders. Therefore, I was able to make conclusions about prosody without using state-of-the-art vocoders.

Finally, in Chapters 3 and 5, I control $F_0$, but not other aspects of prosody. This was motivated by the importance of $F_0$ as an acoustic correlate and the quantity of research on $F_0$ modelling in TTS. Unfortunately, before FastSpeech-2 (Ren et al., 2020), it was not clear how to control acoustic correlates in S2S TTS, thus my use of SPSS models. A WaveNet, driven by linguistic features and acoustic correlates, could have been investigated, but such a model has not been reliably reproduced with data available to academic researchers (van den Oord et al., 2016; Wan et al., 2019).

# Chapter 3

# Diagnosing average prosody through unsupervised control

*This chapter covers the work in "Using generative modelling to produce varied intonation for speech synthesis" (Hodari et al., 2019) presented at the Speech Synthesis Workshop 2019, Vienna, Austria.*

*In this chapter, I investigate "average prosody": flatter and more monotonous intonation. Listening tests provide evidence of average prosody and suggest it is a symptom of modelling assumptions. I conclude that designing a model with control of the prosodic rendition is vital to improving prosody in situations with insufficient context. This is the foundation of my argument involving* ***Theme 1****: we must design controllable voices otherwise they may synthesise average prosody.*

## 3.1 Introduction

Typical TTS systems are designed to synthesise isolated, out-of-context sentences. An **isolated sentence** is a sentence with no additional context. Modelling isolated sentences means prosodic choices in the data are unaccounted for, as the relevant context is missing. This results in models that learn the mean of the prosodic content. Learning the mean is a non-issue for uni-modal data.

However, some aspects of prosody, including intonation, exhibit discrete structure (Silverman et al., 1992; Goodhue et al., 2016; Ward, 2019). Therefore, voices that produce a single prosodic rendition are modelling the mean of a multi-

modal distribution. I demonstrate that this incorrect assumption of uni-modality leads to boring and flat prosody, known as "average prosody". One solution to this failure mode is to expose *control* over this unaccounted-for variation.

In this chapter, I experiment exclusively with intonation, which exhibits discrete structure. The discrete structure of intonation challenges the uni-modal assumption made by most TTS models. To account for this, I focus on modelling the distribution of intonation. This means we can randomly sample any number of renditions for isolated sentences, unlike typical approaches. Given the sampled renditions are meaningfully distinct, the model will, by definition, not produce average prosody. Using these random renditions, I study varied prosody (the opposite of average prosody) for one component of prosody: intonation. I show that controlling for unaccounted-for variation can mitigate average prosody even when there is insufficient context.

Sampling random renditions of isolated sentences may produce inappropriate prosody. In some cases the changes may affect the perceived meaning, in other cases the changes may add variety to the speech without impacting the meaning. One might note that random sampling is uninformed, i.e. it is not context-based. However, as long as renditions are *realistic*, there will exist some context, or multiple contexts, in which they are appropriate—no matter how uncommon a particular rendition might be.[1]

**Realistic** prosody relates to the range of human prosodic behaviour, e.g. could, would, or should a human produce this prosody in any situation? If the answer is no, then the prosody is unrealistic and will not correspond to a real context. A human *could not* produce certain voice qualities, pitch patterns, loudness changes, or other physically limited behaviours. A human *would not* produce prosody that breaks certain rules about stress, declination, syllable structure, etc. A human *should not* produce prosody arbitrarily, e.g. emphasising arbitrary words. I rely on what listeners judge as natural in order to empirically define realistic prosody, since the above definition is not thorough. Given that the prosody is realistic, my investigation of average and varied prosody in isolated sentences can be conducted independently from the question of appropriateness.

To model the distribution of intonation, I train a variational autoencoder

[1]This is supported by evidence in Chapter 5: when context is not given, listeners make an effort to imagine some context in which the prosody would be appropriate.

(Kingma and Welling, 2013) on $F_0$ contours. The variational autoencoder's (VAE) latent variable captures otherwise unaccounted-for variation. Subjective experiments are conducted to determine if average prosody exists in typical TTS systems, if it is resolved by my proposed approach, and if unrealistic prosody has been introduced. I demonstrate that: sampling from low-probability regions of the VAE's prior results in more varied intonation, and that typical TTS approaches produce more average intonation.

## 3.2 Related work

Various methods for unsupervised representation learning for control in TTS have been explored. Watts et al. (2015) proposed sentence-level control using a novel unsupervised approach with discriminant condition codes for TTS—a supervised method originally designed for speaker adaptation in ASR (Xue et al., 2014). Watts et al.'s (2015) sentence-level vectors allow for control of arbitrary variation. While there is a probabilistic interpretation for this approach, Henter et al. (2018b) show it does not model uncertainty in the latent space. As discussed in Section 3.4, uncertainty allows us to determine which renditions are idiosyncratic, and thus more varied.

Global style tokens (GST) are another form of sentence-level control, but using discrete tokens (Wang et al., 2018a). The GST model controls unlabelled variation (i.e. it is unsupervised) and produces high quality speech. However, the model is trained using weighted combinations of individual GST tokens. Synthesising with individual tokens does not produce distinct styles and leads to significantly degraded audio quality.[2] A random weighting of tokens will likely also produce speech with reduced naturalness, since GSTs include no constraints to enforce smoothness on the GST embedding space.

Alternatively, unsupervised representations of prosody can be disentangled from acoustic representations of speech using VAEs (Wan et al., 2019; Wang et al., 2019b). The VAE model presented in this chapter is similar to Wan et al.'s (2019) clockwork hierarchical VAE (CHiVE). However, in addition to $F_0$, CHiVE also models duration and $C_0$ (a correlate of loudness). Wang et al. (2019b) also focus on learning representations of intonation but, like CHiVE, they aim to produce a

[2]This can be observed in the accompanying speech samples for GST (Wang et al., 2018b).

single best rendition. I design TTS systems that can synthesise multiple distinct prosodic renditions in an unsupervised framework and study their behaviour.

## 3.3 Modelling assumptions in TTS

While many methods have been proposed to add control, there is a more fundamental issue, known as oversmoothing, which leads to flatter prosody. Typical neural TTS models are trained using mean squared error (MSE) to predict acoustic features. Optimising MSE is equivalent to minimising the negative log-likelihood of a fixed-variance uni-modal Gaussian. This has two effects on a model: (1) it learns the mean of the data, and (2) it is sensitive to outliers. Along with other modelling assumptions (Henter et al., 2014), this leads to oversmoothing of the acoustic features. In the context of intonation, modelling the mean leads to average prosody.

To mitigate the sensitivity to outliers, methods such as the $\epsilon$-contaminated Gaussian (Zen et al., 2016) can be used. However, a common approach to fix both issues is to collect speech that is as controlled and consistent as possible in terms of style. Training data with a single style results in models which produce more natural speech (Podsiadło and Ungureanu, 2018), but it also limits the voice's stylistic range. To produce more varied style, prosody, or intonation, more varied data is needed, but this additional variation must be handled appropriately by the model.

Generative models, such as Mixture density networks (MDN), have the ability to handle multi-modal data (Bishop, 1994). MDNs parameterise a Gaussian mixture model (GMM) for each acoustic frame which can help with oversmoothing of spectral features (Zen and Senior, 2014). However, for prosodic features, oversmoothing operates over a longer domains, for which frame-level GMMs are less suitable. Instead, I use variational autoencoders, as these can learn an unsupervised representation at whichever domain is preferred, in this case: sentences.

## 3.4 Sampling prosodic renditions using VAEs

Variational autoencoders (VAEs), introduced in Section 2.5.2, are a class of latent variable models, i.e. they learn an unsupervised probabilistic representation

of the data (Kingma and Welling, 2013). They consist of an encoder and a decoder: the encoder parameterises the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$, which is an approximation of $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})$—the underlying factors that describe the data. The decoder is trained to reconstruct the input signal $\mathbf{x}$ from this latent space, i.e. given a sample from the posterior, $\tilde{\mathbf{z}} \sim q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$, the original input is predicted $\hat{\mathbf{x}} \sim p_{\boldsymbol{\theta}}(\mathbf{x} \mid \tilde{\mathbf{z}})$. The encoder and decoder are trained jointly by maximising the evidence lower bound (ELBO),

$$\log p_{\theta}(x) \geq \mathbb{E}_{\tilde{\mathbf{z}} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \tilde{\mathbf{z}})\right] - KL(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \mid\mid p(\mathbf{z})) \tag{3.1}$$

Here I consider conditional VAEs (Sohn et al., 2015), modelling $\mathrm{F}_0$ conditioned on linguistic features. I use a sentence-level approximate posterior. The prior is an isotropic Gaussian, $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{1})$, which gives an analytical form of the KL term.

As an approximate method, variational Bayes—used to derive VAEs—results in trade-offs. If the prior is too simple, as is often the case (Hoffman and Johnson, 2016), the model might prioritise reconstruction when optimising the ELBO, resulting in a latent space that does not match the prior. This mismatch is evidenced by the non-zero KL-divergence between the aggregate posterior and the prior in most VAEs (Dai and Wipf, 2019). While I use a uni-modal prior to model intonation (a multi-modal phenomenon), the latent space can contain multi-modal structure due to this mismatch.[3] This structure is exploited by the tail-based synthesis scheme explained below. The multi-modal structure can be seen in Figure 3.8 (pp. 84).

Enforcing a Gaussian prior gives another useful quality: how close $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ is to the prior mean ($\mu = \mathbf{0}$) will be proportional to how similar $\mathbf{x}$ is to the largest mode in the data (e.g. the most common prosodic style). That is, when encoded, the most average $\mathbf{x}$ will be close to the peak at $\mathbf{z} = \mathbf{0}$, and the most idiosyncratic $\mathbf{x}$ will be far from the peak. This is helpful for generating varied renditions; low-density regions in the prior will correspond to more idiosyncratic renditions. Thus, to study average prosody, I define two models that use only the VAE decoder,

[3]The mismatch between the aggregate posterior and the prior is a failure of VAE's approximate inference. Dai and Wipf (2019) proved the fault can be resolved, though here the mismatch is desirable. As we will see, this exposes the multi-modal nature of intonation.

$$\tilde{\mathbf{z}}_{\text{PEAK}} = \mathbf{0} \qquad \hat{\mathbf{x}}_{\text{PEAK}} \sim p_{\boldsymbol{\theta}}(\mathbf{x} \mid \tilde{\mathbf{z}}_{\text{PEAK}}) \tag{3.2}$$

$$\tilde{\mathbf{z}}_{\text{TAIL}} \sim vMF(\kappa = 0) \qquad \hat{\mathbf{x}}_{\text{TAIL}(r)} \sim p_{\boldsymbol{\theta}}(\mathbf{x} \mid r \times \tilde{\mathbf{z}}_{\text{TAIL}}) \tag{3.3}$$

where $vMF$ is the von Mises-Fisher distribution (Fisher, 1953). For uniform concentration, $\kappa = 0$, this corresponds to a uniform distribution on a hypersphere's surface. $r$ is the radius of this hypersphere.

$\hat{\mathbf{x}}_{\text{PEAK}}$ should correspond to the most common mode, e.g. the most common speaking style or intonation pattern. However, due to the uni-modal prior, $\hat{\mathbf{x}}_{\text{PEAK}}$ may instead correspond to an average of multiple styles, i.e. average prosody. My proposed model uses $\tilde{\mathbf{z}}_{\text{TAIL}}$ to produce idiosyncratic renditions $\hat{\mathbf{x}}_{\text{TAIL}(r)}$, where the larger the radius $r$, the more unlikely the rendition.

## 3.5 Experiments

Average prosody has not been directly measured in the literature. The main aim in this chapter is to provide evidence of average prosody, specifically for intonation. I introduce various systems, including baselines, to enable the design of a listening test that captures the quantity of prosodic variability, i.e. variedness. This listening test can verify if my proposed solution, VAE–TAIL, mitigates average prosody.

### 3.5.1 Systems

Three models were trained: RNN, MDN, and VAE.[4] These are the left three diagrams in Figure 3.1. RNN's and MDN's acoustic models, and VAE's acoustic encoder and decoder all use the same recurrent architecture: a feedforward layer with 256 units, followed by three uni-directional recurrent layers using gated recurrent cells (GRUs) (Cho et al., 2014) with 64 units, finally outputs are projected to the required dimension. Models are implemented in PyTorch (Paszke et al., 2017) using the Morgana TTS toolkit (Hodari, 2020a), and are trained on the Usborne children's audiobook dataset—described in Section 2.4.3.

[4] Code and models are available at github.com/ZackHodari/average_prosody

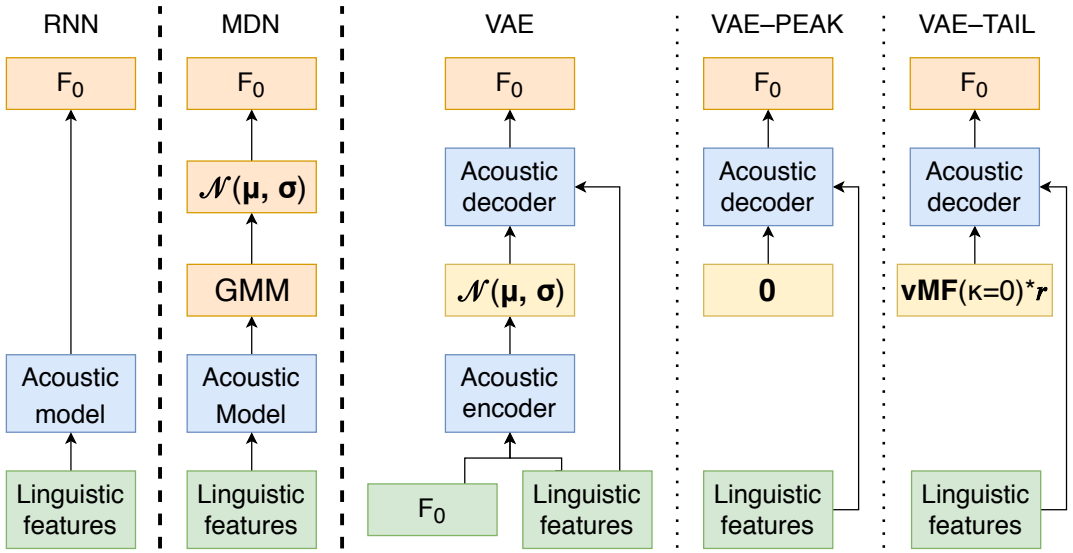



Figure 3.1: $F_0$ models used to assess average prosody. The first three models are trained: RNN, MDN, and VAE. While the last two, VAE–PEAK and VAE–TAIL, are different synthesis-time configurations of VAE. These two systems replace VAE's acoustic encoder with a different prior sampling scheme. Linguistic features and $F_0$ are all at frame level meaning the acoustic models operate at frame level. The yellow latents for the VAE systems are at the sentence level, but are upsampled to frame level meaning that the acoustic decoder also operates at frame level.

The linguistic features consist of 600-dimensional linguistic labels from the standard Unilex (Fitt and Isard, 1999) question-set (detailed in Appendix A) and 9 frame-level positional features, as in the standard Merlin recipe (Wu et al., 2016b). All linguistic features are min-max normalised. The linguistic labels are upsampled to frame-level using natural durations, enabling the model to run at frame-level to predict $F_0$.

The models are trained to predict log $F_0$, delta (velocity), and delta-delta (acceleration) features with mean-variance normalisation. RNN is trained with a mean squared error (MSE) loss, MDN uses negative log-likelihood (NLL), and VAE uses the ELBO defined in Equation 3.1. By modelling only $F_0$, the risk of producing unnatural speech is reduced as spectral features and durations are taken from natural speech. However, it also limits the range of prosodic variation that can be achieved.

Maximum probability parameter generation (Tokuda et al., 2000)—also known as maximum likelihood parameter generation (MLPG)—is used to generate an

$F_0$ contour from the dynamic features. MDN uses predicted standard deviations for MLPG. All other models use the global standard deviation of the training data for MLPG. The WORLD vocoder (Morise et al., 2016) is used for analysis and synthesis.

I use the Adam optimiser (Kingma and Ba, 2014) with an initial learning of 0.005 and a batch size of 32. The learning rate is increased linearly for the first 1000 batches, then decayed proportional to the inverse square of the number of batches (Vaswani et al., 2017, Sec 5.3). Early stopping is used based on validation performance.

MDN has four mixture components, whose variances are floored at $10^{-4}$. This avoids the degenerate case where infinitely peaked modes are learnt. To synthesise from MDN, I use the most likely component sequence (i.e. argmax) to select the means and variances for MLPG. The various choices when synthesising from an MDN are discussed further in Section 3.5.6.

VAE uses a 16-dimensional isotropic Gaussian as the approximate posterior. The latent sample $\tilde{\mathbf{z}}$ is broadcast to frame-level and input to the decoder after concatenating with the linguistic features. The decoder predicts static and dynamic log $F_0$ features, using MSE as the reconstruction loss. VAE's KL-divergence term is weighted by 0 during the first epoch and increased linearly to 0.01 over 40 epochs. Using this annealing schedule, the model converged to a KL-divergence of 3.13. VAE was sensitive to the KL-divergence schedule. Importantly, for the first one or two epochs the weight had to be zero, or very small, otherwise the latent never learns to encode information. Additionally, if the weight goes above some upper bound, matching the prior becomes more beneficial than improving reconstruction, this also leads to posterior collapse, hence the slow annealing schedule.

VAE requires the reference $F_0$ contour as input to the encoder. To synthesise speech from VAE without the reference $F_0$, two sampling schemes are used in place of the encoder: the prior's mean $\tilde{\mathbf{z}}_{\text{PEAK}}$, and low-density samples $\tilde{\mathbf{z}}_{\text{TAIL}}$. The systems VAE–PEAK and VAE–TAIL use these two sampling schemes, respectively. Both systems use the same shared model: VAE's decoder.

To summarise, the 4 systems used to investigate average prosody are:[5]

[5]Speech samples available at zackhodari.github.io/SSW_2019_average_prosody.html

RNN Standard RNN-based SPSS model, using MSE.

MDN MDN with 4 mixture components, using NLL.

VAE–PEAK VAE's decoder using $\hat{\mathbf{x}}_{\text{PEAK}}$ from Equation 3.2, i.e. the latent is the zero vector: $\tilde{\mathbf{z}}_{\text{PEAK}} = \mathbf{0}$.

VAE–TAIL VAE's decoder using $\hat{\mathbf{x}}_{\text{TAIL}(r)}$ from Equation 3.3 with $r = 3$, i.e. the latent is sampled on the surface of a hypersphere with radius 3: $3 \times \tilde{\mathbf{z}}_{\text{TAIL}}$, where $\tilde{\mathbf{z}}_{\text{TAIL}} \sim vMF(\kappa = 0)$.

#### 3.5.1.1 Baselines

To measure average prosody and the presence of unrealistic prosody, we need systems that anchor both aspects. The three baseline $F_0$ systems, used as upper and lower bounds for average prosody and realistic prosody, are:

COPY–SYNTH Natural $F_0$.

BASELINE A quadratic polynomial fitted to the natural $F_0$.

RNN–SCALED $F_0$ from RNN, with the standard deviation scaled by a factor of 3.

BASELINE is designed to serve as a lower bound demonstrating average prosody. Whereas RNN–SCALED is designed as an upper bound exhibiting more varied $F_0$. More varied intonation is only desirable if it doesn't correspond to unrealistic prosody, and if the resulting speech is natural. BASELINE and RNN–SCALED use purposefully simplistic methods to modify $F_0$ variation, this is more likely to result in unrealistic behaviour or unnatural speech. Thus these two system serve as lower anchors on realistic prosody; no viable model should fall beneath them in terms of naturalness. COPY–SYNTH is an upper bound for both axes: it should have varied prosody and realistic prosody.

The amount of perceived variation in RNN–SCALED and VAE–TAIL was calibrated subjectively by matching the level of variation with COPY–SYNTH. The calibration was successful for RNN–SCALED and COPY–SYNTH, but not perfect for VAE–TAIL and COPY–SYNTH, as seen in the results that follow. Performing this calibration automatically or manually is not straightforward, as discussed in Section 3.5.6.1.

### 3.5.2 Evaluation design

I evaluate the amount of variation produced by each system. Specifically, I test the following two hypotheses regarding the variation of $F_0$,

**$H_1$** *Average prosody* — The typical SPSS systems (RNN, VAE–PEAK, and MDN) will have a similar level of **variedness**, but will be less **varied** than COPY–SYNTH and RNN–SCALED.

**$H_2$** *Varied prosody* — VAE–TAIL will be more **varied** than the typical SPSS systems (RNN, VAE–PEAK, MDN).

I use **varied** and **variedness** as antonyms of average or flat. It broadly signifies behaviour that is more dynamic or idiosyncratic. Since I am attempting to categorise which systems produce average or flat $F_0$, my evaluation of variedness (described below) makes direct reference to "flat intonation". In the evaluation "flat intonation" is diametrically opposed with "varied intonation". Thus, variedness is defined by listeners.

However, variation alone is not a guarantee of "better" speech synthesis (Latorre et al., 2014) and unrealistic prosody is especially undesirable as it can result in an uncanny valley phenomenon. For this reason, in addition to measuring variation, I evaluate naturalness. My naturalness listening test uses a MOS design. Naturalness serves as a proxy for signal quality *and* prosody quality, and can indicate unrealistic prosody through low naturalness. I hypothesise that,

**$H_3$** *Realistic prosody* — VAE–TAIL will have similar or better **naturalness** compared to the typical SPSS systems (RNN, VAE–PEAK, MDN).

**$H_4$** *Unrealistic prosody* — RNN–SCALED and BASELINE will be less **natural** than VAE–TAIL and the typical SPSS systems (RNN, VAE–PEAK, MDN).

#### 3.5.2.1 Variedness listening test design

Quantity of prosodic variation, or conversely "prosodic flatness", is a nuanced concept. It would be hard to instruct listeners such that they rate the same concept as one another. Unlike naturalness—a more intuitive concept that represents overall quality—asking "How varied is this speech?" may not result in reliable ratings for a MOS or MUSHRA test (BS Series, 2014).

For a MOS test, the question design is the main challenge. We want listeners to rate on the same perceptual scale. A MUSHRA test can alleviate this since we can require listeners to use the full range of the scale using lower and upper bound systems, e.g. the least varied stimuli is rated 0 and most varied stimuli is rated 100. Thanks to the comparative design, MUSHRA should be more reliable than MOS. However, my evaluation includes 7 systems, this would make for a difficult evaluation task, thus reducing the reliability of a MUSHRA test.

Instead, I use preference (AB) tests to measure variation. Since AB tests directly compare systems they are less noisy than MOS tests, leading to more precise results. Each pairwise comparison is split into a separate AB test, these simpler tasks lead to more reliable, or accurate, results than a MUSHRA test. However, using preference tests comes at the expense of experimental costs.

It is important that listeners focus on the intonation. I ask listeners to choose "which sentence has more varied intonation", where one sentence must be marked as "more flat" and the other as "more varied". The focus on intonation is reinforced by presenting two identical sentences with different renditions side by side. The specific reference to *intonation* is intended to stop listeners choosing based on acoustic quality, which is meant to be captured separately in the naturalness test.

Due to the large number of pairs for 7 systems (21 pairs), BASELINE was excluded from the variedness test (6 systems, 15 pairs). It is clear in the speech samples that it is the least varied (i.e. the most flat). However, this does mean there is no lower-bound on variation in the variedness results.

#### 3.5.2.2 Evaluation setup

Five stories were held out for the listening tests: Hamlet, Pirate Adventures, The Secret Garden, The Story of Cars, and The Story of Chocolate—following the training-validation-test split from Watts et al. (2015). The test stimuli for both listening tests consisted of 32 randomly selected sentences of between 7 and 11 words (1.4 to 4.8 seconds).[6] The listening tests were performed together, with the naturalness MOS test being performed before the variedness AB tests. The tests were performed using a 2x2 Lain square between-subjects design, as each

[6]This range was the inter-quartile range of utterance lengths in the dataset.

sentence had 22 screens: 7 systems in the MOS test and 15 pairs for the AB tests. In total 30 listeners completed the test—15 per listener group in the Latin square. The test was ~45 minutes long and listeners were paid £8.

### 3.5.3 Naturalness results

A summary of the naturalness ratings is provided in Figure 3.2. A Wilcoxon rank-sums significance test between all pairs of systems in the naturalness test was conducted, followed by Holm-Bonferroni correction—the same statistical analysis as for the Blizzard challenge (Clark et al., 2007). VAE–TAIL, RNN, MDN, and VAE–PEAK form a group within which no significant differences were found. All other system pairs are significantly different, with a corrected p-value of less than 0.00001.

### 3.5.4 Variedness results

While it is not guaranteed that human preferences are self-consistent, or globally consistent,[7] the variedness results in Figure 3.3 do form a consistent ordering from most flat to most varied:

RNN → VAE–PEAK → MDN → VAE–TAIL → COPY–SYNTH → RNN–SCALED

However, relative variedness is sometimes inconsistent, e.g. while RNN–SCALED is more varied than COPY–SYNTH ($5^{\text{th}}$ row), the difference between COPY–SYNTH and RNN ($13^{\text{th}}$ row) is greater than the difference between RNN–SCALED and RNN ($15^{\text{th}}$ row). This shows that the subjective ratings produce a non-linear scale along the axis of variedness.

A binomial significance test for the 15 pairs in the listening test was performed, followed by Holm-Bonferroni correction. With multi-test correction the $1^{\text{st}}$ row (VAE–PEAK, RNN), $2^{\text{nd}}$ row (MDN, VAE–PEAK), and $5^{\text{th}}$ row (RNN–SCALED, COPY–SYNTH) did not show a significant difference ($p > 0.05$): this is indicated by the lighter colouring of those pairs in Figure 3.3. All other pairs are significantly different, with a corrected p-value of less than 0.0002.

[7] As described by Arrow's impossibility theorem (Arrow, 1950).

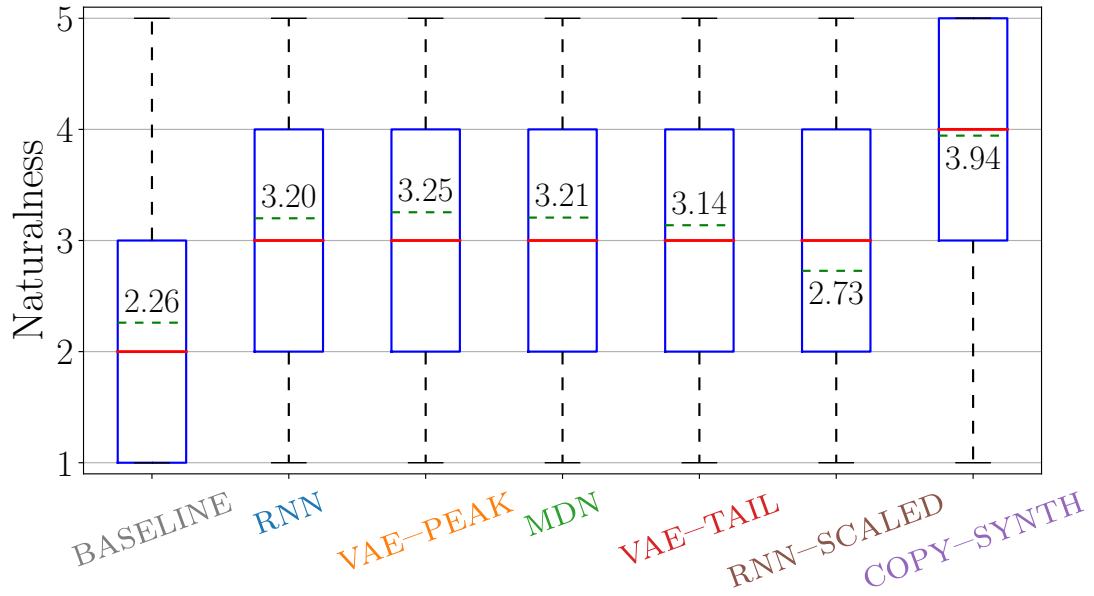



Figure 3.2: Naturalness results. Solid red lines are medians, dashed green lines are means, blue boxes show the 25th and 75th percentiles, and whiskers show the range of the ratings, excluding outliers (of which there are none). Systems are ordered according to the variation test.

#### 3.5.4.1 Interpreting pairwise preference results

While the ordering of variedness matches the expected order in hypotheses $\mathbf{H_1}$ and $\mathbf{H_2}$, it is not immediately clear if the hypotheses are supported, as relative variedness is inconsistent. For example, system A can be 50% better than system C (i.e. $A \gg C$), and system B 25% better than system C (i.e. $B > C$), but A could be 25% *worse* than system B (i.e. $A < B$). If we represent the 15 pairwise preference tests in Figure 3.3 on a single dimension of relative variedness, it will be possible to compare all systems.

Multi-dimensional scaling (MDS) could be used to summarise the pairwise results (Borg and Groenen, 2003). However, MDS treats comparisons as distances, whereas the pairwise preferences correspond to directed edges. Instead, I formulate the problem as a system of linear equations,

$$A\mathbf{x} = \mathbf{b} \tag{3.4}$$

where $A \in \{-1, 0, 1\}^{15\times 6}$, $\mathbf{x} \in \mathbb{R}^{6\times 1}$, and $\mathbf{b} \in \mathbb{R}^{15\times 1}$. The linear equations $A_i$ encode two things for each pairwise test $i$: which two systems ($n_{\uparrow}^{(i)}$, $n_{\downarrow}^{(i)}$) were

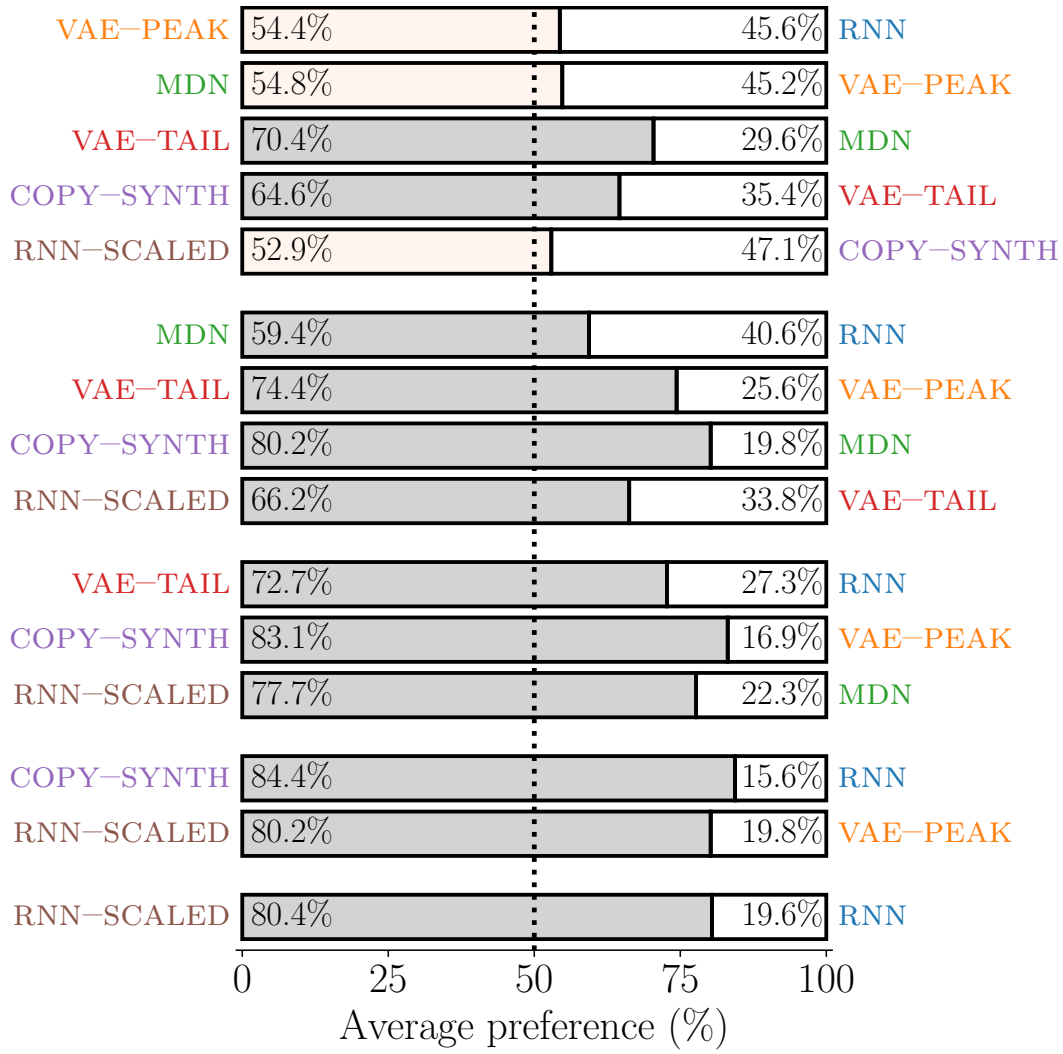



Figure 3.3: Pairwise variedness results. Pairs are ordered such that the more varied system is on the left. The top 5 rows give the pairs that are consecutive in the ordering, with following rows showing systems that are increasingly further apart in the ordering. No significant difference was found for pairs marked in a lighter colour.

part of pair $i$, and which of these systems were more varied ($n_{\uparrow}^{(i)}$) and less varied ($n_{\downarrow}^{(i)}$). $x_n$ is the unknown value for system $n$ that we want to find, as described below this is the relative variedness. $b_i$ represents how much better one system was than another in pairwise test $i$.

For each system pair ($n_{\uparrow}^{(i)}$, $n_{\downarrow}^{(i)}$) the equation $A_i\mathbf{x} = b_i$ contains only two non-zero coefficients in $A_i$: $+1$ for the more varied system ($n_{\uparrow}^{(i)}$) and $-1$ for the less varied system ($n_{\downarrow}^{(i)}$). The constant term, $b_i$, describes the "excess preference" of a system pair. Excess preference is defined as the number of percentage points between the preferences in a single AB test,

$$b_i = y_{\uparrow}^{(i)} - y_{\downarrow}^{(i)} \tag{3.5}$$

where $y_{\uparrow}^{(i)}$ and $y_{\downarrow}^{(i)}$ are the pairwise preference results for system pair $i$, i.e. the percentages of row $i$ in Figure 3.3. Thus, the equation for the $i^{\text{th}}$ pair, $A_i\mathbf{x} = b_i$, takes the following form,

$$(+1)x_{n_{\uparrow}^{(i)}} + (-1)x_{n_{\downarrow}^{(i)}} = y_{\uparrow}^{(i)} - y_{\downarrow}^{(i)} \tag{3.6}$$

By defining this system of linear equations, the unknown variables $\mathbf{x}$ will represent the preference of each system when taking into account the relative comparisons in all other pairwise tests. That is, $\mathbf{x}$ will represent the dimension of relative variedness.

This system of linear equations is inconsistent and overdetermined: inconsistent because the rank of the coefficient matrix is different from the rank of its augmented matrix, and overdetermined because there are more equations (pairs) than unknowns (position of systems). While no exact solution exists for such a system, ordinary least squares can be used to find a solution with minimal error as follows,

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b} \tag{3.7}$$

The solution, $\mathbf{x}$, is plotted in Figure 3.4. Since the scale of the solution is arbitrary, no units are given.[8] Systems to the left have flatter intonation and systems to the right have more varied intonation. This axis represents human preference and is not intended to be a perceptual scale. Interestingly, this

[8]Only the distance between points in Figure 3.4 is meaningful. For example, $\mathbf{x} + 10$ is also a valid solution.


Figure 3.4: Relative variedness derived by solving a system of linear equations representing the 15 pairwise preference results in Figure 3.3.

mathematically unbiased transformation places COPY–SYNTH as the most varied system, not RNN–SCALED. While the difference is insignificant, it illustrates how relative variedness can be inconsistent; COPY–SYNTH was more varied than more systems than RNN–SCALED was, even though RNN–SCALED was slightly more varied than COPY–SYNTH in a direct comparison ($5^{\text{th}}$ row of Figure 3.3).[9]

### 3.5.5 Naturalness–variedness trade-off

The relative variedness results allow us to evaluate how the systems compare in terms of average or varied prosody. In order to consider any trade-offs relating to the production of unrealistic prosody or unnatural speech, relative variedness (Figure 3.4) is plotted against mean naturalness (Figure 3.2) in Figure 3.5. BASELINE was excluded from the variedness AB tests, so it is represented as a horizontal line with no known value for relative variedness. Since BASELINE was clearly less varied than all other systems in informal listening, it would be placed some distance to the left of RNN in Figure 3.5. Though how far this distance would be is unknown.

The clearest result seen in Figure 3.5 is the clustering among the typical SPSS systems (RNN, VAE–PEAK, MDN). These all have similar naturalness and are much less varied than the other systems. This supports $\mathbf{H_1}$, providing evidence that average prosody exists in typical SPSS systems. Whereas VAE–TAIL is much more varied, suggesting that it does not suffer from average prosody, at least not to the same extent, supporting $\mathbf{H_2}$.

It is not possible to bound the axis of variedness as BASELINE was excluded from the variation tests. However, the proximity of VAE–TAIL to COPY–SYNTH

[9]The slight difference between RNN–SCALED and COPY–SYNTH is insignificant. This comparison is merely intended to illustrate how the proposed method deals with inconsistent pairwise results.

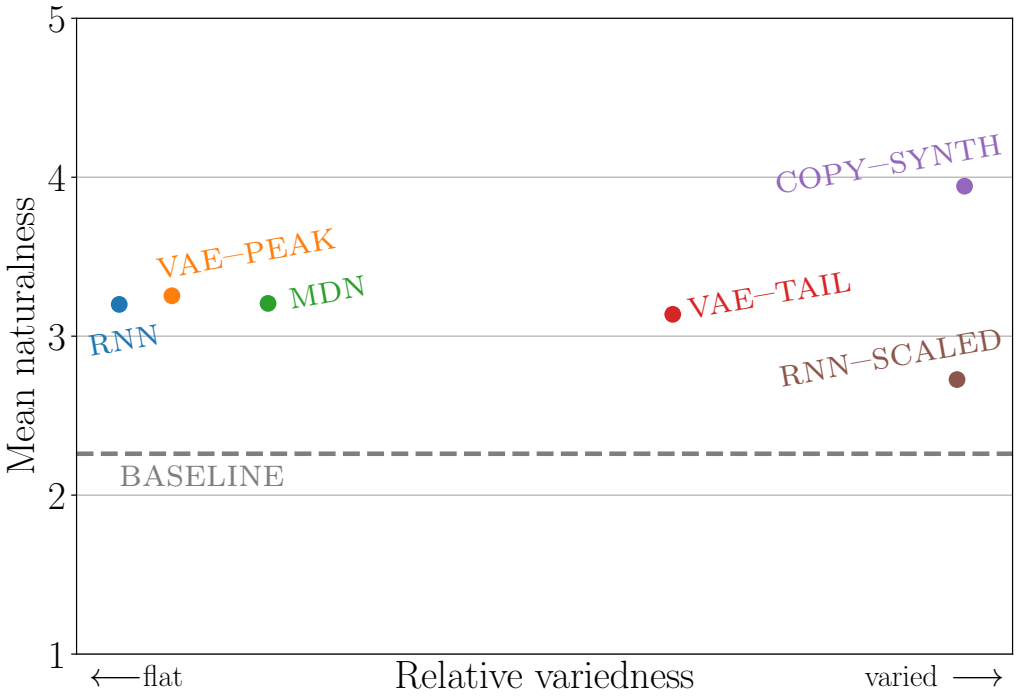



Figure 3.5: Naturalness-variedness trade-off. Ideally as the amount of prosodic variation is increased VAE–TAIL will not decrease in naturalness. Direct comparisons between any systems on either axis should only be made using the significance results

and RNN–SCALED suggests that it produces varied prosody, further supporting $\mathbf{H_2}$. Importantly, no significant difference in naturalness was found between VAE–TAIL and the typical SPSS systems, this provides evidence for $\mathbf{H_3}$: VAE–TAIL does not produce unrealistic prosody.

RNN–SCALED and BASELINE produce significantly less natural speech, meaning they produce either unrealistic prosody or unnatural speech, supporting $\mathbf{H_4}$. This suggests the listening test was correctly designed—it captured the lower-bounds as having lower overall quality. The test also successfully measures different amounts of variation independently of naturalness (e.g. VAE–PEAK and VAE–TAIL have the same level of naturalness, but very different relative variedness), and measures the same amount variation for systems with drastically different prosodic and acoustic quality (e.g. COPY–SYNTH is significantly more natural than RNN–SCALED, but both have the same level of variedness).

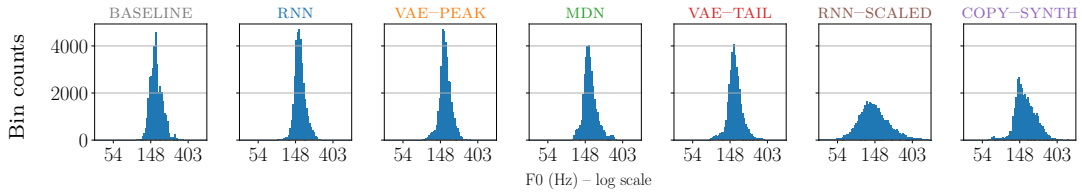



Figure 3.6: Histogram of log $F_0$ values for each system over all the listening test material. Ordered according to the variation test.

## 3.5.6 Analysis

### 3.5.6.1 Calibration

Perceptual phenomena such as naturalness are very challenging to measure objectively (Lo et al., 2019). The same is true for measuring the amount of $F_0$ variation. This was problematic when calibrating the synthesis hyperparameters: the radius, $r$, for VAE–TAIL; and the scale factor for RNN–SCALED.

In calibrating the radius and scale factor, the aim was to match the amount of perceived variation with COPY–SYNTH. Initially, I used the standard deviation of the $F_0$ contours for different settings of the synthesis hyperparameters. Histograms of $F_0$ predictions in Figure 3.6 illustrate this objective variation metric for the 7 proposed systems. However, this objective metric ($F_0$ standard deviation) did not align with my perception of quantity of variation.

The mismatch between the objective and perceived variation can be illustrated using the variedness listening test results. A comparison between objective variation—standard deviation of $F_0$ predictions (Figure 3.6)—and the perceived variation detailed by the relative variedness results (Figure 3.4) is presented in Figure 3.7. MDN and VAE–TAIL appear to be similar according to the objective metric, however there is a large disparity according to the subjective results. Conversely, RNN–SCALED seems to be more objectively varied compared with COPY–SYNTH, but listeners thought they were equally varied. These disparities show that this objective metric is misleading for calibration of perceived variation. For this reason, VAE–TAIL was calibrated by hand to match COPY–SYNTH. As seen in Figure 3.4, $r$ could have been increased further.

**Probability mass in vae–tail** Further research into sampling from the prior from Byrne (2021, Section 4.3.1) showed that: due to the curse of dimensionality,

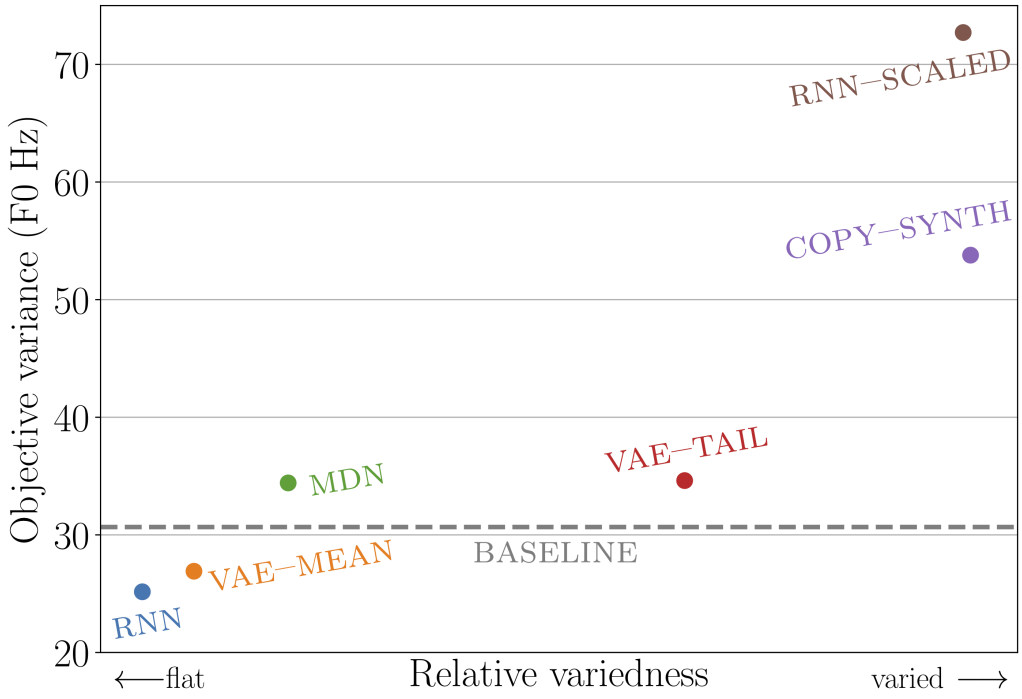



Figure 3.7: Subjective variedness vs. objective variation. Subjective variedness comes from human preference results in Figure 3.4. Objective variation is the standard deviation of $F_0$ predictions, illustrated in Figure 3.6.

even for a standard Gaussian, the majority of probability mass lies outside the hypersphere with $r = 3$ for 16 dimensions. This means that the tail sampling approach presented here does not in fact index on idiosyncratic renditions. However, through the calibration process discussed above, the sampling did produce perceivable variation, albeit more common $F_0$ variation.

#### 3.5.6.2 Multiple renditions

I argued that a system capable of producing multiple distinct prosodic renditions would not produce average prosody: this motivated the design of VAE–TAIL. While the listening tests provided evidence of varied intonation, I have not demonstrated if VAE–TAIL can produce multiple renditions, or how renditions differ from each other.

Figure 3.8 illustrates the range of possible $F_0$ contours produced by VAE–TAIL for a single sentence. This density plot was created by synthesising 10,000 $F_0$ contours, $\hat{\mathbf{x}}_{\text{TAIL}(3)}$, using samples $\tilde{\mathbf{z}}_{\text{TAIL}} \sim vMF(\kappa = 0)$. Thanks to the use

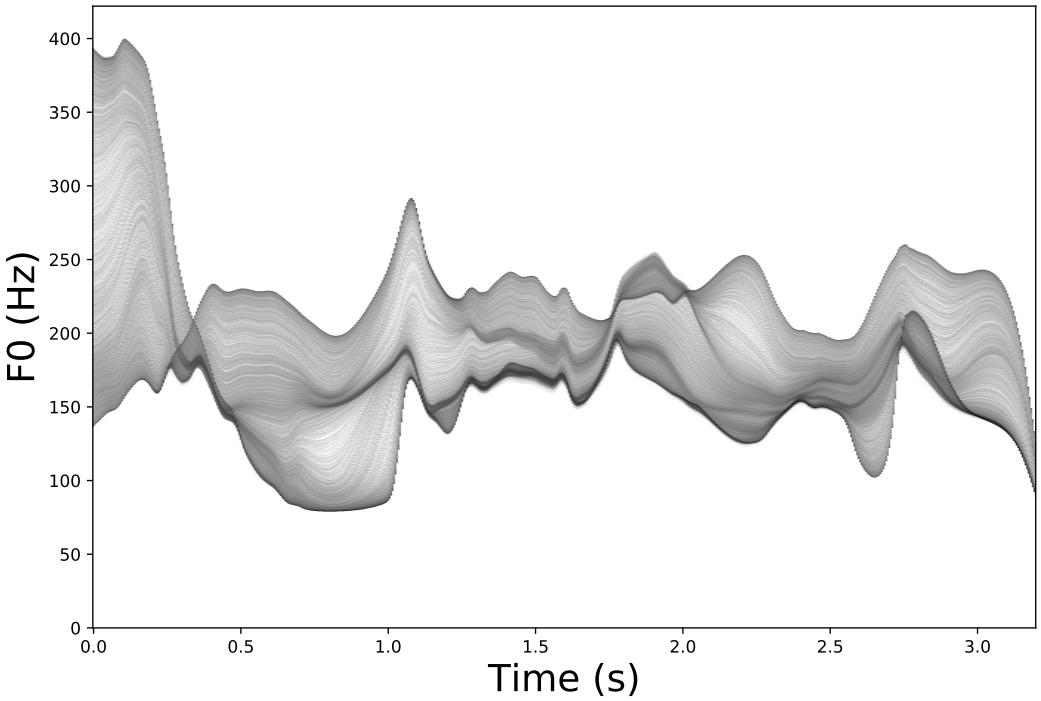



Figure 3.8: Density of $F_0$ predictions made by VAE–TAIL for the sentence "Goldilocks skipped around a corner and saw..." Darker regions indicate a cluster of samples that produce a similar $F_0$ contour, i.e. a mode.

of a smooth prior in training, $\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{1})$, there is continuous variation between different $F_0$ contours. This corresponds to moving across the surface of the hypersphere that $\mathbf{z}_{\text{TAIL}}$ samples from. More importantly, there are denser (i.e. darker) collections of $F_0$ contours, these clusters correspond to modes. These modes are different distinct contours that can be controlled by the latent space even without context. In Chapter 5, I look specifically at the distinctiveness of renditions in a similar VAE model.

The presence of modes in Figure 3.8 when sampling symmetrically from the prior suggests that the latent space learnt by the VAE is not a standard normal distribution. This is also corroborated by the non-zero KL divergence of the VAE's aggregate posterior. Typical SPSS systems produce average prosody as they assume a uni-modal distribution. While the VAE uses a uni-modal prior, Figure 3.8 demonstrates that, due to the approximate nature of variational inference, the VAE is not limited to learning uni-modal structure.[10]

[10]This could be seen as a failure of the graphical model, though by design if a VAE did perfectly match the prior, the latent variable would be useless. In this case, we could consider

#### 3.5.6.3 MDN sampling

Like VAEs, MDNs are generative models. However, sampling from the frame-level GMMs in an MDN is not straightforward. Though MLPG can be used to select the single best trajectory (Tokuda et al., 2000, Case 3), producing multiple renditions requires a sampling strategy.

Randomly choosing mixture components for each frame produced noisy $F_0$ contours. Alternatively, using a single component for the entire sequence and comparing to other components did not reveal distinct renditions. Both of these sampling approaches likely failed for the same underlying reason: the components don't represent modes of $F_0$ variation, but instead capture frame-level variation. This may, in part, be because mixture components behave in a similar way to the $\epsilon$-contaminated Gaussian distribution (Zen et al., 2016): they use mixture components to describe outliers. Ultimately, MDN is unable to produce significant variation as it models uncertainty at the frame level, whereas the VAE captures uncertainty at the sentence domain.

## 3.6 Conclusion

This chapter investigated the modelling assumptions made by typical SPSS systems. I presented two new TTS systems using a VAE: VAE–PEAK was designed to make the same modelling assumptions as typical SPSS models, and VAE–TAIL was designed to address these assumptions. Through a novel evaluation design I provided the first direct evidence of average prosody—flatter, less varied intonation—and demonstrated that this is due to a modelling assumption where unaccounted-for variation is ignored. My evaluation also demonstrated that designing a model which produces more varied prosody did not lead to the generation of unrealistic prosody.

By comparing VAE–TAIL to an MDN model, I was also able to demonstrate the importance of the domain at which prosody variation is captured. There still exist issues with using the sentence domain for capturing prosody. In Chapters 5 and 6, I explore phrase and word level prosody representations for improved prosodic representation learning.

---

the mismatch between the prior and the aggregate posterior as "a feature not a bug".

The pairwise results visualised in Figure 3.4 could also be analysed using pairwise rank aggregation (Bradley and Terry, 1952) to statistically rank the systems. While pairwise rank aggregation does not calculate positions on an axis like the approach taken here, it does produce an ordering and associated significance results for consecutive systems in this ordering. This analysis could be combined with the existing analysis to provide significance grouping within the axis of relative variedness.

My results demonstrate the importance of **Theme 1** (controllability): without control of unaccounted-for variation, like in VAE–TAIL, typical SPSS models produce average prosody. This provides support for a portion of the thesis's claim: *"...prosodic variation not determined by the available context must be controlled..."* The following two chapters focus on two contrasting approaches for achieving interpretable control (**Theme 2**).

# Chapter 4

# Interpretable control of variation without human annotation

*This chapter covers the work in "Learning interpretable control dimensions for speech synthesis by using external data" (Hodari et al., 2018) presented at Interspeech 2018, Hyderabad, India.*

*As demonstrated in Chapter 3, unaccounted-for prosodic variation in speech must be controlled. Control must either be predicted using context information, or directed by a human-in-the-loop. If there is insufficient context, we must rely on human control. However, to make such systems more usable, the control mechanisms must be interpretable (**Theme 2**) to human users. I investigate a method of control using human-defined labels, but without human annotation of the TTS data—which is expensive. This is achieved using an additional non-TTS dataset and pseudo-labelling. When evaluating a human-in-the-loop system using my approach, I find that listeners prefer intentionally controlled speech over randomly varying speech, reinforcing that prosodic choices must be made appropriately.*

## 4.1 Introduction

Most speech synthesis research focuses on non-controllable TTS, but as demonstrated in Chapter 3, it is important that TTS systems can produce multiple prosodic renditions. In order for representations to be useful for human-in-the-loop control of prosody they must have some level of interpretability. This can

also prove useful with context-based prosody prediction, for debugging and analysis purposes.

For a control mechanism to be **interpretable** the prosodic effects it controls must relate to some abstract concepts in the mind of the human using the system. These effects could relate to: prosodic context information, such as emotion, attitude, or sentiment; prosodic phenomena such as prominence, phrase breaks, or voice quality; or, acoustic correlates of prosody, such as $F_0$, intensity, or durations. Whichever abstract concepts are controlled, human operators must understand what the effect will be on the resulting prosody.

The most straightforward approach to capture human-understood concepts is to rely on human nomenclature, e.g. human annotations. This is a common method for controllable TTS. Labels are collected to describe the variation of interest, such as, emotion (Douglas-Cowie et al., 2003), emphasis (Cole et al., 2017), speaking style (Wood and Merritt, 2018), or prosodic structure (Silverman et al., 1992). Alternatively, the variation of interest may be elicited (Busso et al., 2008; Goodhue et al., 2016; Prateek et al., 2019), in which case labels exist by design. However, annotating or eliciting variation is expensive, especially for large high quality TTS datasets. While existing labelled data may exist, it is unlikely to be high enough quality for current TTS technology.

In this chapter, I make use of found TTS data—the Usborne children's audiobook dataset discussed in Section 2.4.3. Found data that is produced with genuine communicative intent, such as audiobooks (King et al., 2018; Zen et al., 2019) and podcasts (Székely et al., 2019), includes interesting variation typically avoided when recording TTS data. While found data is also unlikely to be labelled, it is high enough quality for statistical parametric speech synthesis (SPSS) (King et al., 2018).

To account for the lack of labels, I propose the use of an additional external dataset that contains human annotations of the variation to be controlled. The primary TTS dataset is automatically annotated using the labelling schema of the external dataset using pseudo labelling (Lee, 2013). This enables a TTS model to be trained using the primary TTS data and controlled using the external dataset's labelling schema. I demonstrate this approach for emotion control. The primary synthesis dataset (Usborne) must exhibit the variation described by the emotion

labels. Since the external dataset is not directly used for training a TTS model, its acoustic quality does not need to be as high, it also does not need to be transcribed.

## 4.2 Related work

### 4.2.1 Emotion annotation

External emotional expression is a paralinguistic function and is conveyed, in part, through prosody, e.g. in pitch, speaking rate, and loudness changes (Vinciarelli et al., 2009). True emotional state is internal. Only through a person's external expression—conscious and unconscious choices of language, prosody, and body language—can their internal emotional state be interpreted by others (Picard and Picard, 1997). For facial expressions, emotion is interpreted consistently across cultures (Ekman et al., 1987). Evidence for this consistency in speech has also been found (Pell et al., 2009). The inaccessibility of any internal emotional state underlies the difficulty of annotating emotion, as with many other context factors that impact prosodic choices.

Various paradigms for describing emotion have been proposed. "Pure emotions" is a psychological theory suggesting only one emotion can be portrayed at a time (Plutchik, 1984). Annotations based on this theory are known as "categorical emotions" (Ekman, 1992). However, this schema does not account for the intensity of emotions (Ekman et al., 1987). Additionally, it is rare to observe pure emotions in natural speech (Cowie and Cornelius, 2003).

A practical solution to mitigate these drawbacks is to relax the "pure" nature of categorical emotions. This can either be achieved using multiple annotators (Busso et al., 2008), by annotating with a weighted combination of pure emotions (Mower et al., 2011), or both.

However, a set of basic emotions must still be selected for this paradigm. There is no agreement on such a set of emotions (Douglas-Cowie et al., 2003). The most common set includes: happy, sad, angry, and neutral (Lee and Tashev, 2015; Kim et al., 2013; Lee et al., 2011). This clearly excludes a wide range of emotional expression. From a machine learning research viewpoint, such a crude schema might be necessary in the earlier stages of research.

Appraisal-based emotions, or "dimensional emotions", are an alternative paradigm to categorical emotions. Dimensional emotion annotation is motivated by cognitive theory (Lazarus, 1991). Dimensional emotions define a set of traits relevant to emotion expression. Lazarus (1991) proposed: arousal, a measure of activeness; and valence, a measure of positivity. Fontaine et al. (2007) demonstrated that two dimensions are insufficient to capture emotion and suggested adding: dominance, a measure of control; and expectancy, a measure of predictability. It is common to use only three dimensions: arousal, valence, and dominance (Busso et al., 2008).

Annotation of dimensional emotions is not without issues. While it is much more descriptive than categorical emotions, annotating dimensional emotions is a much more complex task. More complex tasks typically lead to less accurate annotator behaviour and lower inter-annotator agreement. In the emotion dataset described in Section 4.4.1, inter-annotator agreement is measured with an average Cronbach's alpha of 0.67 (Busso et al., 2008)—the reliability of these labels may be questionable (Cortina, 1993).

### 4.2.2 Emotion recognition

Emotion recognition can be a regression or classification task for both annotation schemas: categorical and dimensional emotions. Categorical emotions can be "soft" if using multiple annotators or weighted annotation. Conversely, dimensional emotions can be discretised into categories. As such, the range of methods for emotion recognition is broad.

At the time of conducting this research, there was no dominant machine learning approach—except perhaps neural networks in general. Different modelling approaches are summarised in Table 4.1 (pp. 99). A lot of focus was typically placed on the input features.[1] Hand-engineered features were dominant, as were brute-force exhaustive feature sets containing thousands of hand-engineered sentence-level features (Valstar et al., 2014; Schuller et al., 2016). Following the 2015 Audio/Visual Emotion Challenge (AVEC), which used a smaller set of 102 features, the field shifted away from the brute-force feature set approach (Ringeval et al., 2015). In particular, Eyben et al. (2016) demonstrated that these exhaus-

[1] Research also focused heavily on the correct design of datasets (Douglas-Cowie et al., 2003), including the design of annotation schemas.

tive feature sets were unnecessary, and that a minimal set of, at most, 88 features could achieve state-of-the-art performance. In this chapter, I use Eyben et al.'s (2016) minimal feature set: the extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS). eGeMAPS features were adopted for AVEC 2016 (Valstar et al., 2016).

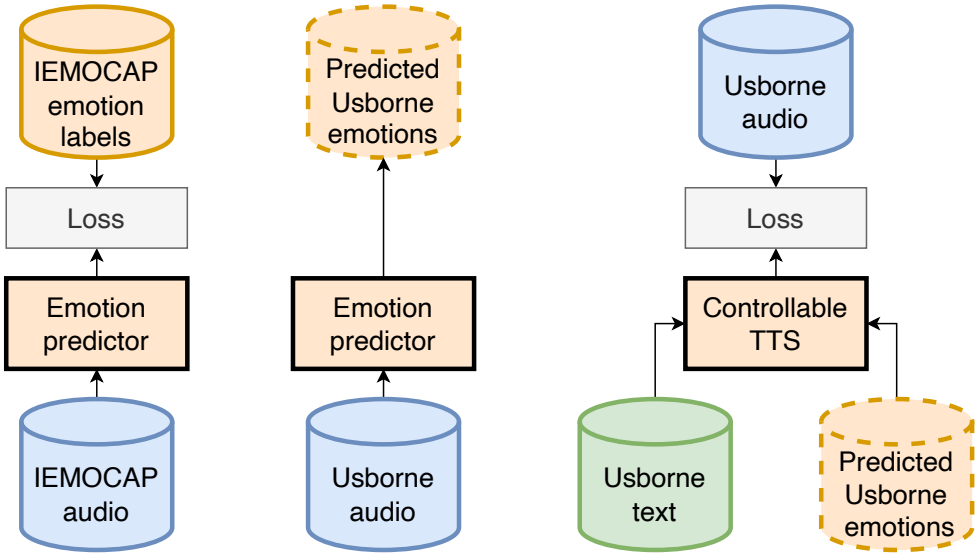
Research had also began to use raw input, as opposed to hand-crafted features. Trigeorgis et al. (2016) used only the waveform as input, achieving significantly better performance than state-of-the-art. Ghosh et al. (2016) used the spectrogram as input, resulting in increased discrimination between happy and angry. However, at the time of this work, such methods were not widespread.

## 4.3 Emotive TTS using pseudo labels

To create a controllable TTS model, I use additional features—also known as input codes (Luong et al., 2017)—in an SPSS model. However, the Usborne TTS data used to train the voice consists of *(text, audio)* tuples, with no additional labels. While the phone sequence is augmented with other linguistic features automatically extracted from the text, these additional linguistic features are typically limited to syntactic information and some prosodic structure information. Instead, my approach uses a secondary external speech dataset to train a recognition model: an emotion predictor. This model is used to augment the primary TTS dataset with labels, or *pseudo labels*. This can be thought of as an offline version of pseudo-labelling (Lee, 2013; Arazo et al., 2020).

Given the predicted pseudo labels for the primary TTS dataset, a controllable SPSS model can be trained. This approach generalises to different types of variation exhibited in speech. As discussed, I demonstrate the method for emotion control. The stages of the training process are illustrated in Figure 4.1: (a) train an emotion predictor on external non-TTS data (Section 4.3.1); (b) extract pseudo labels using the emotion predictor; and (c) train the TTS model on the primary TTS data using the predicted emotions (Section 4.3.2).

At synthesis time there are two methods for selecting control inputs. First, when a natural reference rendition of the target emotion exists we can use the emotion predictor to extract the emotion and transfer these control values. Second, if no reference exists for a target emotion, the control values can be manually




(a) Training of emotion predictor

(b) Pseudo-labelling of TTS dataset

(c) Training of controllable TTS model

Figure 4.1: Stages of training a controllable TTS model using an external dataset to label the primary TTS dataset. (a) The emotion predictor is trained using the IEMOCAP emotion dataset. (b) The trained emotion predictor is used to predict emotions for the Usborne TTS data that are used as labels, this process is referred to as pseudo-labelling. (c) The controllable TTS model is trained using the Usborne TTS dataset and the predicted emotion pseudo-labels.

specified: this human-in-the-loop control use-case is detailed in Section 4.3.3. A third synthesis method could be added: use additional information to predict the emotion, this third approach is explored in Chapter 6.

### 4.3.1 Emotion predictor

As discussed, there are two dominant emotion annotation approaches (Douglas-Cowie et al., 2003): categorical, and dimensional. Dimensional emotions suffer from lower inter-annotator agreement, and it is harder to evaluate their interpretability in subjective tests. Therefore, I use categorical emotions as the pseudo labels. Since there is useful information in the dimensional labels and they already exist in the emotion dataset used, I predict these as a secondary task when training the recognition model.

My emotion prediction model classifies categorical emotions (e.g. happy), and uses multi-task learning (Caruana, 1998) to predict dimensional emotions

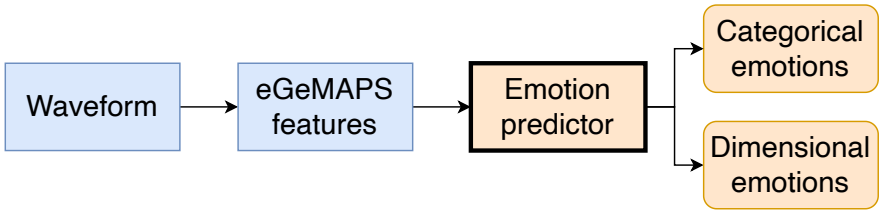



Figure 4.2: Emotion predictor, trained using IEMOCAP data. Used to label Usborne TTS dataset with categorical emotion predictions.

(e.g. level of arousal) as a secondary task. The emotion predictor is illustrated in Figure 4.2, the model uses 3 feed-forward layers: a shared layer of 200 units, two parallel private layers of 20 units, and projections to the target dimension for the classification task (4 dimensions) and regression task (3 dimensions). The model was trained using categorical cross-entropy for classification, and binary cross-entropy for regression. This architecture was the best performing model in previous work (Hodari, 2017a).

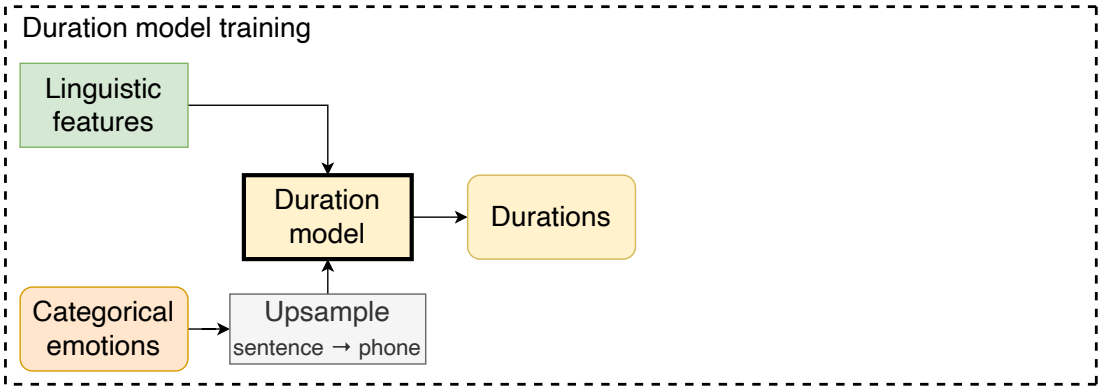
I use the eGeMAPS features as inputs (Eyben et al., 2016). eGeMAPS features are utterance-level statistics of low-level descriptors (LLDs). LLDs are temporal features and include: frame-level energy, spectral, cepstral, prosodic, and voicing descriptors. A full account of the LLDs can be found in Appendix A. The features are designed to be predictive of emotions and are chosen for their ability to model perceptually relevant changes in speech.

### 4.3.2 Controllable SPSS model

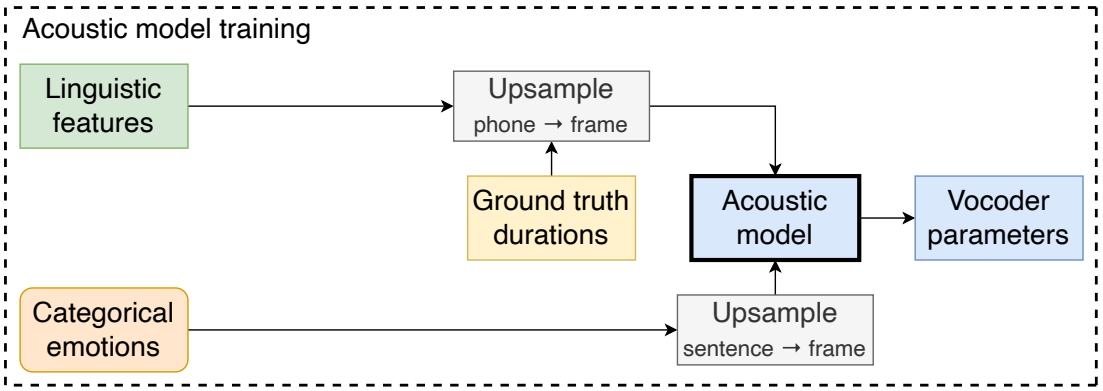
The TTS model is an SPSS model. Both the duration and acoustic models are adapted using control inputs predicted by the emotion predictor, i.e. the pseudo labels. This TTS model can control any aspect of speech described by the vocoder features, unlike the systems explored in Chapter 3 which only varied $F_0$.

The duration and acoustic models (Figure 4.3) consist of 6 tanh layers with 1024 hidden units, following the *fls_blizzard2017* recipe in Merlin[2] (Wu et al., 2016b). The duration model (Figure 4.3a) is trained using phone-level linguistic features—detailed in Appendix A—and emotion inputs from the emotion predictor, upsampled to phone-level by repetition. To train the acoustic model (Figure 4.3b), natural durations are used to upsample the linguistic features and

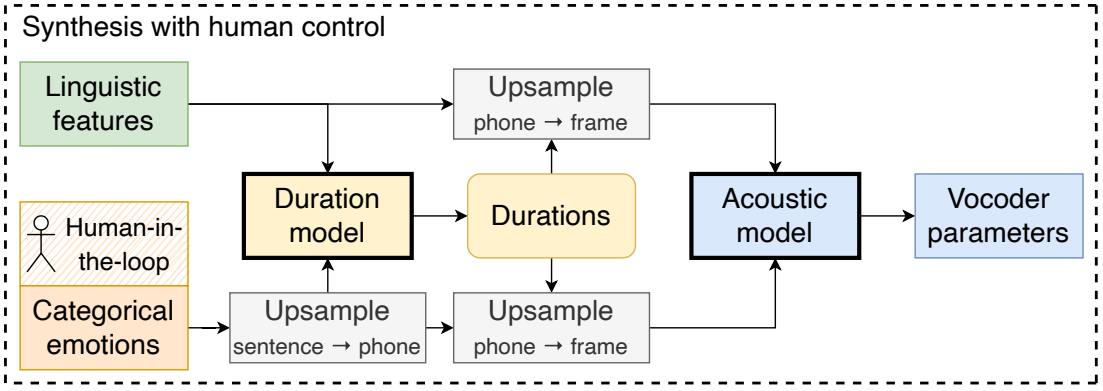
[2]Available at github.com/CSTR-Edinburgh/merlin/tree/master/egs/fls_blizzard2017




(a) Duration model training with emotion control.




(b) Acoustic model training with emotion control.




(c) Synthesis with the controllable TTS model, using a human-in-the-loop to choose the categorical emotions. The UI in Figure 4.4 is operated by the human-in-the-loop to control the TTS model. Alternatively, the emotion can be copied from a reference waveform using the emotion predictor (not illustrated).

Figure 4.3: Controllable TTS model training and human-in-the-loop synthesis.

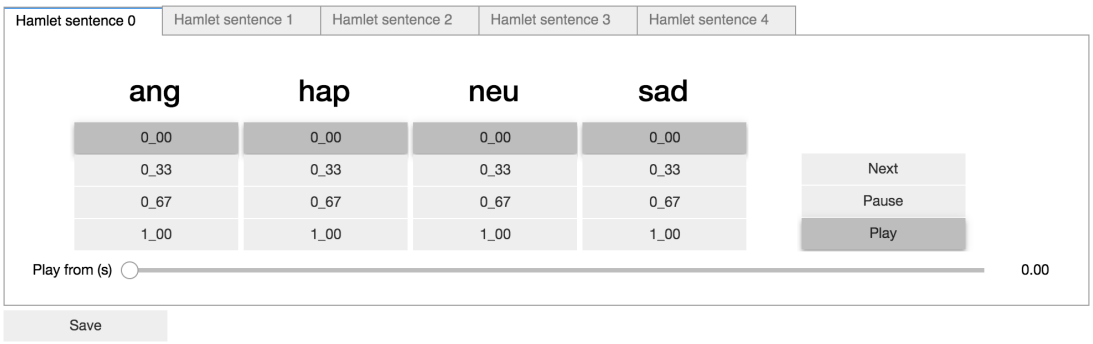



Figure 4.4: UI used by human-in-the-loop to choose control values.

emotion inputs to frame-level by repetition. At synthesis time (Figure 4.3c), durations predicted by the duration model are used to upsample the linguistic features and emotion inputs.

As per the *fls_blizzard2017* Merlin recipe, linguistic features are min-max normalised, and durations and vocoder features are mean-variance normalised. The acoustic model predicts vocoder features: F0, smoothed spectrogram, and aperiodic energy, along with all delta and delta-delta features. MLPG is used to find the most probable trajectory from these dynamic acoustic feature predictions (Tokuda et al., 2000). The trajectory from MLPG is used by STRAIGHT to synthesise the waveform (Kawahara, 2006).

### 4.3.3 Human-in-the-loop control

As discussed there are two methods for selecting control inputs: using a reference audio signal, and using human-in-the-loop control. For the human-controlled use-case I designed a simple graphical interface, shown in Figure 4.4. The interface is designed for 1 or more contiguous utterances in an extract (e.g. a paragraph). The emotion of each utterance can be modified and the extract can be played in sequence. With this interface, lexical and prosodic content of previous and future utterances can be used as context by the human-in-the-loop when choosing the rendition for a given utterance. I evaluate the perception of stimuli created by a human-in-the-loop in Section 4.4.4.2. To create the stimuli for this listening test, I operated the control interface in Figure 4.4.

## 4.4 Experiments

In this chapter, I aim to demonstrate that TTS with interpretable, human-defined control is possible without expensive annotation of TTS data. In Section 4.4.4 I evaluate the interpretability of the proposed controllable TTS system. After verifying the efficacy of my approach, I evaluate the appropriateness of long-form speech created by a human-in-the-loop. But first, I introduce the data used (Section 4.4.1), and discuss objective results for both emotion recognition (Section 4.4.2) and TTS (Section 4.4.3).

### 4.4.1 Datasets

As discussed, two datasets are used: a primary TTS dataset, and a secondary dataset with human-defined labels used for training the emotion predictor. Both of these must exhibit the variation to be controlled, that is, the variation described by the emotion labels.

#### 4.4.1.1 Emotion recognition database

The *Interactive Emotional Dyadic Motion Capture* dataset (IEMOCAP) is an acted dialogue dataset (Busso et al., 2008). IEMOCAP contains 12.5 hours of data from both scripted and improvised sessions between two actors. There are 5 male and 5 female actors. Mixed-gender dyads were recorded for two sessions of roughly 1 hour. Each session contains an average of 15 conversations. Scripted dialogues were designed to produce one of 5 *pure* emotions—anger, sadness, happiness, frustration, and "neutral". Improvised conversations used hypothetical scenarios designed to elicit these 5 emotions.

Each utterance is an average of 4.5 seconds, and includes labels from 3 annotators for both categorical and dimensional emotion—arousal, valence, and dominance. The use of multiple annotators allows for soft emotion labels to be considered in place of pure emotions. Soft labels correspond to the average of all annotators' responses. This mitigates some issues with pure emotions, since it defines a probability distribution over multiple pure emotions. Additionally, the probability densities should correlate with emotion intensity: the more annotators that agree on one emotion the more salient that emotion must be, and vice versa.

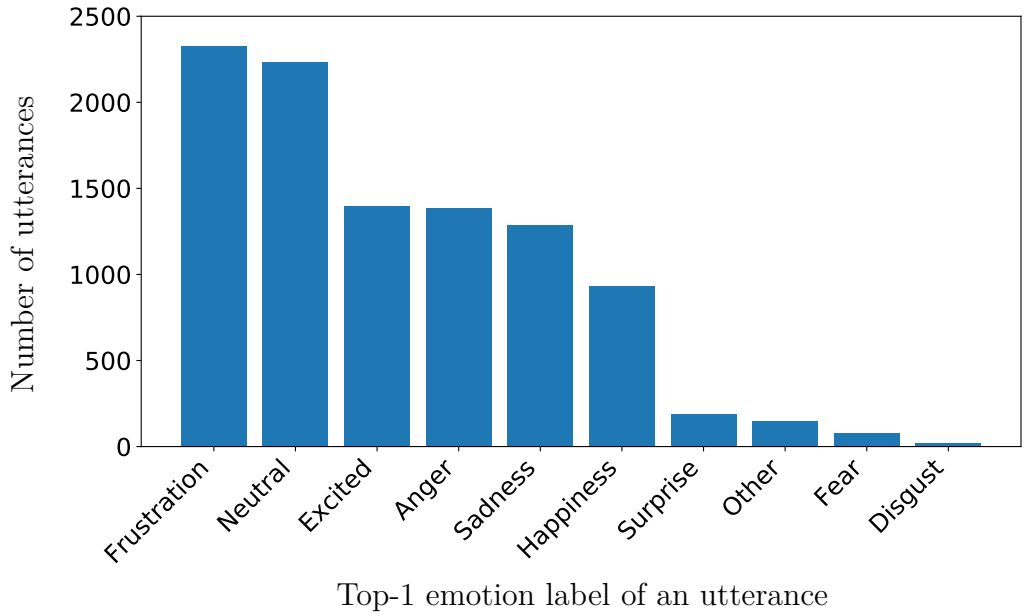



Figure 4.5: Frequency of top-1 (i.e. argmax) emotion labels for IEMOCAP data.

IEMOCAP's categorical emotion annotations include the 5 categorical emotions the script was designed to elicit, as well as: disgust, fear, excitement, surprise, and "other", for a total of 10 categories. By adding new categories after data collection, the curators of IEMOCAP reduced the constraints on actors' performances and mitigated the need for improvised scenarios to accurately elicit the intended emotion. This should make the labelling schema more representative of the data collected.

Emotion classification is a noisy task, for humans and machine learning models alike (Douglas-Cowie et al., 2003). Human judgements are noisy, in part, due to the varying implicit definitions held in annotators' minds. One approach to mitigate this is using a subset of emotions. I use a subset of IEMOCAP consisting of happy, sad, angry, and neutral utterances only. This is common for research using IEMOCAP due to the class imbalance between the 10 categories, seen in Figure 4.5.[3]

[3]While frustration is the most common emotion in Figure 4.5, it is commonly omitted in the literature, as illustrated by the examples in Table 4.1. This is likely due to the tradition of considering a common set of four "basic" emotions: happy, sad, angry, and neutral. I chose to use the same basic four emotions, to enable comparison with results from related work.

#### 4.4.1.2 TTS database

The primary TTS dataset is the Usborne children's audiobook dataset, described in Section 2.4.3. This found data includes interesting prosodic variation and expressive speech. Five stories were held out for the listening tests: Hamlet, Pirate Adventures, The Secret Garden, The Story of Cars, and The Story of Chocolate—following the training-validation-test split in Watts et al. (2015).

### 4.4.2 Emotion recognition

The emotion predictor was trained using modNN (Hodari, 2017b), a wrapper around Tensorflow (Abadi et al., 2016).[4] The model maps from utterance-level eGeMAPS features to utterance-level labels: 4 "soft" categorical emotions, happy, sad, angry and neutral; and 3 dimensional emotions, arousal, valence, and dominance.

The recognition data was split into 5 cross-validation folds, using 4 dyads for the training set in each fold. The evaluation set, containing 1 dyad per fold, is split in half for validation and testing. Similar to Lee and Tashev (2015), I use 1 evaluation set speaker for hyperparameter tuning during validation, and the other is held-out as the unseen test set and used only when reporting recognition results. To avoid the held-out test set consisting of a single gender when aggregated across cross-validation folds, the held-out speaker's gender was alternated in each fold. Early stopping was used based on the validation set. All results presented use the held-out test speakers.

#### 4.4.2.1 Recognition results

The emotion predictor (Figure 4.2, pp. 93) achieved an emotion classification accuracy of 62.9%. Multiple architectures were tested, but performance ceilinged at around 62% with the feed-forward architecture. Convolutional and recurrent architectures, based on the LLD features, were attempted but with less success. Early stopping was helpful for achieving better accuracy on the validation set. Overfitting was observed without early stopping.

To put the performance of 62.9% in context, Table 4.1 reports comparable results from the literature. All listed works use IEMOCAP and predict the same 4 emotions.

[4] Code and models are available at github.com/ZackHodari/IS18_control_space.

Table 4.1: Overview of comparable IEMOCAP recognition results, classifying the same emotions: angry, happy, sad, and neutral.

| **Method** | **Input features** | **Accuracy** |
|---|---|---|
| Feed-back long short-term memory (LSTM), attention (Zhang et al., 2017) | Predicts using previous emotion | 60.8% |
| Ensemble support vector machine (Rozgic et al., 2012) | 12 MFCCs, jitter, shimmer | 60.9% |
| Convolutional neural network, multiple kernel learning (Poria et al., 2016) | ComParE 2016 (Schuller et al., 2016) | 61.3% |
| Deep belief network, support vector machine (Xia and Liu, 2015) | ComParE 2010 (Schuller et al., 2010) | 62.5% |
| **Feed-forward neural network** (Hodari, 2017a) (*proposed model*) | **eGeMAPS** (Eyben et al., 2016) | **62.9%** |
| Recurrent neural network, extreme learning machine (Lee and Tashev, 2015) | MFCCs, $F_0$, voice probability, zero-crossing rate | 63.9% |
| Progressive deep neural networks (Gideon et al., 2017) | eGeMAPS (Eyben et al., 2016) | 65.7% |
| Convolutional neural network, LSTM (Satt et al., 2017) | Spectrogram (cropped to 3 seconds) | 68.8% |

#### 4.4.2.2 Labelling the TTS data

The Usborne TTS dataset lacks emotion labels. The emotion predictor is used to label this data with *pseudo labels*. This corresponds to domain transfer from the multi-speaker IEMOCAP dialogue data to the single-speaker Usborne audiobook data.

The distribution of predictions on the Usborne data is shown in Figure 4.6. Neutral is predicted as the most probable emotion most often, with happy being predicted as most probable the least often. Figure 4.7 shows the top-1 emotions, both for labels in IEMOCAP (a subset of the data in Figure 4.5) and predictions on Usborne (representing the same data as in Figure 4.6). The bias towards neutral, and against happy in the IEMOCAP data is mirrored in the predictions

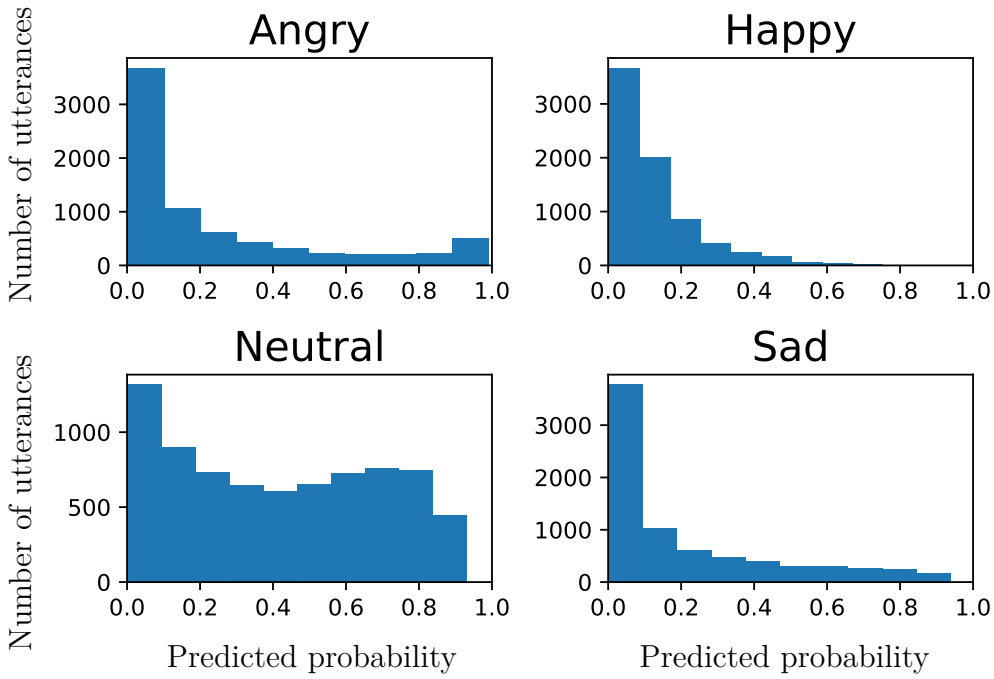
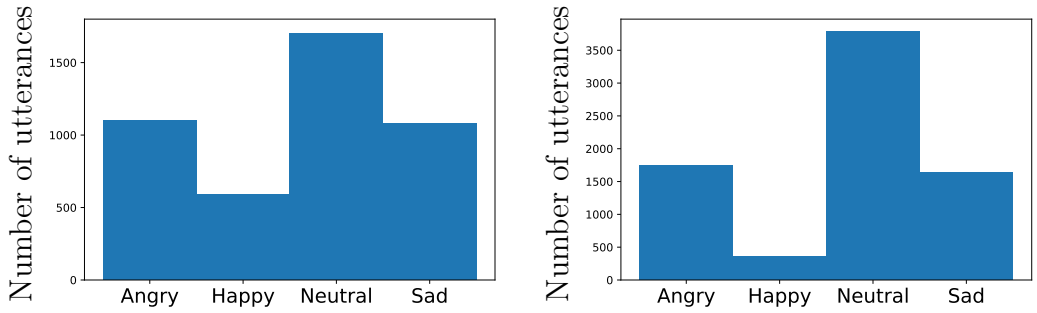



Figure 4.6: Histograms of predictions of each emotion's probability on the Usborne data for categorical emotions.




(a) Top-1 labels for IEMOCAP data. (b) Top-1 predictions for Usborne data.

Figure 4.7: Histogram of top-1 (i.e. argmax) emotion categories for IEMOCAP (annotated labels) and Usborne (predicted pseudo-labels).

on Usborne. The distribution of predictions on the Usborne data could be the true distribution, it could be caused by the bias in the IEMOCAP data, or it could be related to domain transfer issues caused by style mismatch between the two datasets.

### 4.4.3 Controllable SPSS

The speech synthesis systems were trained using the open-source Merlin toolkit (Wu et al., 2016b). Two systems are evaluated here:

*DNN-B* — Baseline SPSS voice, with no control.

*DNN-C* — Proposed emotion controllable SPSS voice (Figure 4.3, pp. 94).

A second version of *DNN-C* controlled by dimensional, as opposed to categorical, emotion labels was trained. However, designing subjective evaluations for dimensional emotion control is more challenging. As such, this second version of *DNN-C* was not evaluated. Without proper evaluation it would be misleading to comment on the interpretability of the control inputs. Though the voice did train successfully, and the control inputs did impact the resulting speech.

#### 4.4.3.1 Objective results

While objective metrics do not always correlate with subjective measures, they can be useful for validating training, or flagging issues. In Table 4.2, I report 4 metrics commonly used for objective evaluation of SPSS when predicting vocoder parameters. The metrics are defined as follows, where $y_t$ and $\hat{y}_t$ represent the target and predicted vocoder parameter at frame $t$, respectively:

- **Mel-cepstral distortion (MCD):** The average of each frame's Frobenius norm for mel-spectrograms. The first dimension, $y_{t,1}$, is excluded as this is $C_0$ and represents loudness.

$$\text{MCD} = \frac{10\sqrt{2}}{\ln 10}\frac{1}{T}\sum_{t=1}^{T}\sqrt{\sum_{d=2}^{D}(y_{t,d}-\hat{y}_{t,d})^2}$$

- **Band aperiodicity distortion (BAP):** The average of each frame's Frobenius norm for aperiodic energy bands.

$$\text{BAP} = \frac{1}{10}\frac{1}{T}\sum_{t=1}^{T}\sqrt{\sum_{d=1}^{D}(y_{t,d}-\hat{y}_{t,d})^2}$$

Table 4.2: Objective performance of SPSS with and without control vectors.

| | **Objective metric** | | | |
|---|---|---|---|---|
| | MCD (dB) | BAP (dB) | $F_0^{(RMSE)}$ (Hz) | VUV (error %) |
| DNN-B (baseline) | **5.650** | **0.075** | 51.209 | **7.451** |
| DNN-C (with reference control) | 5.719 | 0.076 | **50.624** | 7.551 |

- **log $F_0$ error:** Root mean squared error (RMSE) for voiced frames that were correctly predicted as voiced (i.e. true positives). $y_t^{(vuv)}$ and $\hat{y}_t^{(vuv)}$ are the target and predicted voiced/unvoiced status of the $t^{th}$ frame.

$$F_0^{(RMSE)} = \sqrt{\frac{1}{|T_{voiced}|} \sum_{t \in T_{voiced}} (y_t - \hat{y}_t)^2}$$
$$\text{where } T_{voiced} = \{t \in \{1, \ldots, T\} \mid y_t^{(vuv)} \wedge \hat{y}_t^{(vuv)}\}$$

- **Voiced/unvoiced error (VUV):** Percentage of frames with incorrect voicing decision (i.e. $1 - accuracy$).

$$VUV = \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}(y_t^{(vuv)} \neq \hat{y}_t^{(vuv)})$$

The objective results in Table 4.2 show that, when using the reference emotion (i.e. predicting using eGeMAPS features from the reference audio), *DNN-C* is worse for cepstral features and voiced-unvoiced prediction, this suggests quality might be slightly reduced. Interestingly, *DNN-C* performs better for $F_0$ prediction, this is likely because the eGeMAPS features used to predict the reference emotion label are derived from the waveform. Specifically, eGeMAPS features include sentence-level $F_0$ information (cf. Table A.2, pp. 169).

We cannot measure objective quality for other control inputs as there are no additional reference renditions for individual sentences. In Figure 4.8, I illustrate the qualitative variation between various manually-specified control inputs. The plots show duration and $F_0$ outputs for 4 control input settings, each representing one of the 4 categorical emotions. The spectral feature predictions also vary, but are not included as they are less visually interpretable. While the duration and $F_0$ variation looks subtle, Figure 4.8 demonstrates that the model does respond

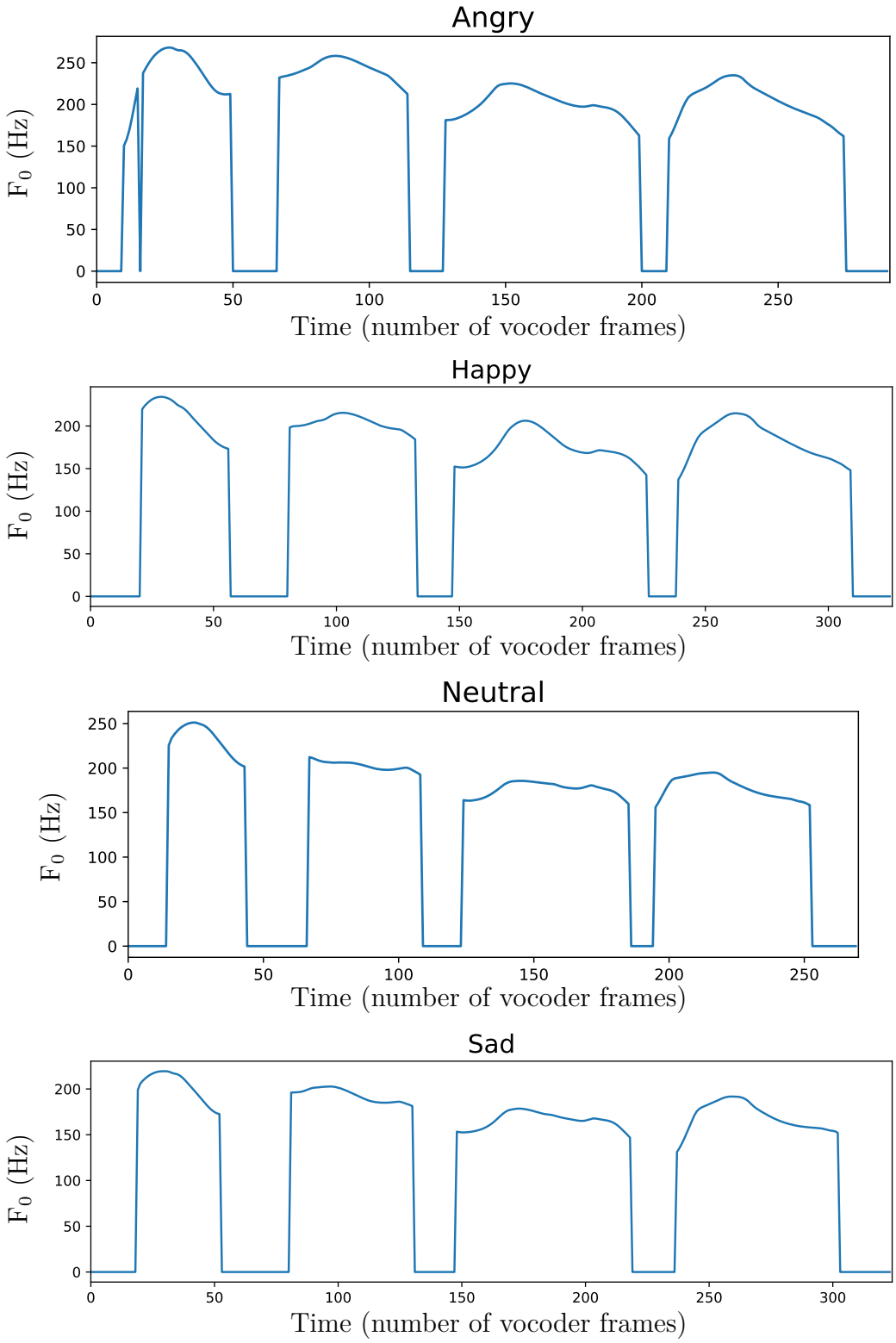



Figure 4.8: Demonstration of duration and $F_0$ variation produced by *DNN-C* as the control vector is changed.

to changes in the control values. The listening tests evaluate this formally, finding that listeners do perceive this variation as conveying different emotions.

### 4.4.4 Subjective evaluation

Two main questions are investigated using human listeners: (1) is the learnt control mechanism interpretable, and (2) can a human control the voice to produce more appropriate prosody. I conduct two listening tests to answer these questions. I test interpretability with respect to human-defined categorical emotions in Section 4.4.4.1. I explore listener preference in long-form reading for intentional or "irregular" human control in Section 4.4.4.2. 31 native English-speaking university students (26 female, 5 male) were paid to carry out these two tests in sound-proof booths using using Beyerdynamic DT770 headphones. The tests were implemented using BeaqleJS (Kraft and Zölzer, 2014).[5] Together, the two tests took roughly 1 hour, for which participants were paid £10.[6]

#### 4.4.4.1 Interpretability

The emotion labels used to train *DNN-C* are predictions (as opposed to human annotations), therefore they will be noisy and may be subject to biases in the IEMOCAP data and the emotion predictor. We must verify that *DNN-C* accurately produces the intended emotions. Participants were asked to label synthesised sentences according to the 4 categorical emotions: happy, sad, angry, and neutral. By presenting stimuli that aim to convey a certain emotion, a positive result will demonstrate if the control mechanism is interpretable.

The test material consists of 50 audiobook sentences, each performed with 4 different styles. The 50 sentences are a random subset of the Blizzard Challenge 2017 test set (King et al., 2017). The 4 renditions of each sentence correspond to the 4 categorical emotions—the renditions in Figure 4.8 were prepared in the same way. These 50 sentences were not controlled for linguistic content, which may bias listeners towards certain responses. However, it should be possible to overcome this bias if the model's control of emotion is strong enough, since the sentences are presented as isolated utterances to listeners.

[5] Code is available online, github.com/zackhodari/beaqlejs. My fork includes additional features and bug fixes.

[6] Speech samples are available at zackhodari.github.io/IS18_control_space.html

Table 4.3: Confusion matrix for the forced-choice emotion classification task. Stimuli are generated by *DNN-C* with control inputs set to one of four one-hot vectors, each representing a different categorical emotion. The accuracy for each intended emotion is in bold face.

| Correct class | Predicted class | | | |
|---|---|---|---|---|
| | Angry | Happy | Neutral | Sad |
| Angry | **30%** | 51% | 13% | 7% |
| Happy | 36% | **13%** | 29% | 22% |
| Neutral | 10% | 15% | **66%** | 10% |
| Sad | 10% | 4% | 30% | **56%** |
| | MEAN ACCURACY **41%** | | | |

Participants performed a forced-choice labelling task, selecting the closest emotion from the 4 categorical emotions for each of the 200 utterances. Allowing users to provide free-form responses instead may lead to more accurate results, but this would make analysis more challenging—something explored in Chapter 5.

Table 4.3 presents a confusion matrix of participant responses. Average classification accuracy is 41% (chance level is 25%). To place the result of 41% accuracy in context, the inter-annotator accuracy in IEMOCAP is 48% for the 4 emotions used in this chapter. Annotator agreement in IEMOCAP is likely to be higher as listeners were labelling natural speech, whereas in this listening test the stimuli are synthetic speech. Similarly, for natural speech, Banse and Scherer (1996) conclude that human performance for emotion labelling is around 50%—this was not linked to a particular number of emotion categories. They also cite two studies evaluating human performance for a 5-class emotion classification task of natural speech, reporting accuracies of 64% (Bezooijen, 1984), and 56% (Scherer et al., 1991). With this context—that human agreement is low for emotion annotation—the performance of 41% for *DNN-C* is satisfactory. Thus, the controllable voice, *DNN-C*, is able to modify perceived emotion according to human-interpretable categories.

Looking at the confusion matrix (Table 4.3) in more detail, *happy* is only recognised correctly 13% of the time, and 36% of *happy* renditions are incorrectly perceived as *angry*. Similarly, 51% of *angry* renditions are incorrectly perceived

as *happy*. This poor performance is likely a symptom of imperfect labelling from the emotion predictor, which is related to the class imbalance in the IEMOCAP labels (Figure 4.7a, pp. 100). The emotion predictor is biased against predicting *happy* (Figure 4.7b, pp. 100), which may lead to compounding errors such as this. In addition, it is known that *happy* and *angry* are difficult to distinguish between using acoustic features (Klabbers et al., 2007). Since the system is able to perform emotion control with 41% accuracy, I did not investigate this further. While there is a mismatch in style between the Usborne and IEMOCAP datasets, these interpretability results demonstrate that there is sufficient overlap.

#### 4.4.4.2 Human-in-the-loop control

Prosody control has two clear use-cases: human-in-the-loop control, and context-based (i.e. automated) control. Here, I look at listener perception of human-controlled speech, comparing it to the default style and to irregular variation (i.e. inappropriate variation). Evaluation is performed using paragraphs of more than one sentence, this allows listeners to account for context when determining their preferences. Comparing *DNN-C* to the default style, created using conventional uncontrolled SPSS, will determine if a human-in-the-loop can add appropriate prosody using an interpretable control mechanism. On the other hand, comparing *DNN-C* to irregular variation will provide information on the role of the human-in-the-loop, and on listener preference relating to appropriate prosody. Irregular variation is achieved using random sampling, meaning it is not determined by the context.

Three systems are compared in this test:

*DNN-B* — Baseline SPSS voice, with no control.

*DNN-C* — Proposed emotion controllable SPSS voice (Figure 4.3, pp. 94). Here a human-in-the-loop controls the sentence-by-sentence variation through the control interface in Figure 4.4 (pp. 95).

*DNN-R* — Random control of *DNN-C*. This system controls *DNN-C* with randomly chosen emotion values, instead of using a human-in-the-loop to choose the control inputs intentionally. This irregular behaviour will mostly produce inappropriate prosody.

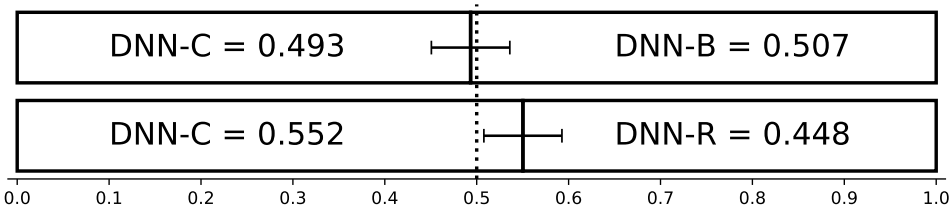



Figure 4.9: Evaluation results of human-in-the-loop control for paragraphs, including 95% confidence intervals. This evaluation compares the human-controlled *DNN-C* system to a non-controllable baseline *DNN-B* and a system with inappropriate variation *DNN-R*.

Two pairwise preference tests are conducted: *DNN-C* vs. *DNN-B*, and *DNN-C* vs. *DNN-R*. The former comparison will provide similar conclusions to a comparative MOS test, demonstrating if listeners prefer default or human-controlled prosody, i.e. can a human-in-the-loop add more appropriate prosody. The latter comparison uses the same model for both systems, differing only in the prosodic choices. A significant results for this test would demonstrate that listeners can perceive prosodic choices as more or less appropriate, ideally any significant result would favour *DNN-C* as it uses human-controlled prosody.

System pair *DNN-B* and *DNN-R* was excluded due to the length of the test—this listening test was done jointly with the interpretability test from the previous section. This pairwise comparison may have provided information about average prosody vs. inappropriate prosody. However, in this chapter I focus on controllability and interpretability, not average prosody, hence these conclusions are less relevant.

The test material consisted of 17 short audiobook paragraphs, from the Blizzard Challenge 2017 test set. The paragraphs had an average duration of 24 seconds when synthesised. For *DNN-C*, the human-in-the-loop process of finding satisfactory renditions for each sentence took between 2 and 3 minutes per paragraph using the UI and was conducted by me. Listeners were presented with two versions of the same paragraph (from two different systems) and asked to "choose the paragraph you would prefer if you were listening to an audiobook for pleasure". By linking the task to the audiobook domain of the TTS data, this should provide ratings of a paragraph's appropriateness. As usual, this is likely to be entangled with assessment of acoustic quality.

Preference ratios and 95% binomial proportion confidence intervals for the two pairwise tests are presented in Figure 4.9. A binomial significance test found no significant difference between the first pair: *DNN-C* and *DNN-B*. This suggests the proposed system with a human director does not degrade the quality, but also that it has not improved the prosody. While improving the prosody's appropriateness would be ideal, this is still a positive result as we have enabled control of the speech without sacrificing quality. It is possible that due to the relatively short nature of the paragraphs used, there was not enough time for listeners to perceive and assess the sentence-to-sentence variation. For the second pair: *DNN-C* and *DNN-R*, the proposed system with appropriate control, *DNN-C*, is significantly preferred. This demonstrates that listeners prefer variation that fits the text and context, i.e. appropriate prosody.

## 4.5 Conclusion

I have demonstrated that interpretable control can be added to a TTS system without expensive annotation of the training data. This was achieved using a secondary external speech dataset containing human annotations of the variation to be controlled, in this case: emotion. The proposed system is interpretable with respect to the labels used in the external dataset (**Theme 2**), this makes control operated by a human-in-the-loop more time efficient.

This method can be used to control other types of variation in speaking style. Given existing labelled speech data, labels can be transferred to the primary TTS dataset through pseudo-labelling. This does require the TTS dataset to exhibit the variation labelled in the external dataset. The external labelled speech data does not need to be transcribed, and it does not need to be recorded to the same quality standards as the primary TTS dataset. My approach uses transfer learning and must contend with domain mismatch between training and pseudo-labelling. Cai et al. (2020) extended the work presented here using maximum mean discrepancy to adapt the emotion predictor in an unsupervised fashion, thus mitigating domain mismatch through fine-tuning.

In long-form reading, i.e. for paragraphs, listeners significantly preferred prosody chosen based on the context, over prosody chosen randomly. While this preference may be intuitive, it explicitly motivates the pursuit of producing appropriate prosody, either through human control or automated prediction.

In the following chapter, I look at another method to achieve interpretable control of prosody (**Theme 2**). Instead of relying on human-defined schemas during training, interpretable control is pursued using inductive biases in an unsupervised approach.

# Chapter 5

# Perception of discrete representations for prosodic control

*This chapter covers the work in "Perception of prosodic variation for speech synthesis using an unsupervised discrete representation of F0" (Hodari et al., 2020) presented at Speech Prosody 2020, Tokyo, Japan.*

*In this chapter, I further investigate **Theme 2**, looking at the interpretability of learnt representations for prosody control (**Theme 1**). Specifically, I explore the perceived effects of unsupervised representations, since there is no clear understanding of what they control—in part because this is difficult and expensive to evaluate. The perceptual study I propose provides a means to explore unlabelled representations. Understanding what prosodic behaviour is controlled by a representation is also useful for **Theme 3**, as this knowledge can inform what context information is most relevant to the prosodic behaviour being controlled.*

*An unsupervised representation of intonation is learnt using a variational autoencoder extended from Chapter 3. I choose to learn discrete representations as these are more human-usable and easier to evaluate in terms of interpretability. Additionally, the representation is learnt at the prosodic phrase domain. Compared to typical utterance-domain representations, the phrase domain is more suited for prosody modelling.*

## 5.1 Introduction

Natural speech contains prosodic variation caused by many different contextual factors. Enabling control of this variation is important to avoid average prosody and to support selection or prediction of more appropriate prosody. Using learnt representations, we can produce varied prosodic renditions, as seen in Chapter 3. However, it is difficult to predict control parameters due to the lack of context. Unfortunately, increasing the available context is non-trivial; relevant context can be very broad, and expensive or impractical to obtain.

Understanding what categories of prosodic variation a learnt representation exposes would inform which context would be most impactful if available. If collected, such context can be used to better predict prosodically appropriate representations. Previous work has identified consistent variation in prosody with respect to: discourse structure (Farrús et al., 2016; Cole and Reichel, 2016; Kleinhans et al., 2017; Aubin et al., 2019), information structure (Calhoun, 2010; Lai, 2012a), speaker attitude (Armstrong and Prieto, 2015; Gravano et al., 2008a; Lai, 2010; Betz et al., 2019), speaker stance (Hübscher et al., 2018; Freeman, 2019; Ward et al., 2018), and personality traits (Bawden et al., 2016). Clearly, the context required for each of these categories can be very different.

Recent phonetic studies support the idea that both categorical and continuous features are integral to prosodic variation (Grice et al., 2017; Cole et al., 2017). In line with these observations, I learn representations with *multi-modal* structure. This structure should capture both: categorical differences, such as those associated with phrasing and prominence; and fine-grained phonetic differences which can vary the perception of expressivity, emphasis, and speaker affect.

In order to understand what prosodic behaviour the learnt representation captures, I conduct a qualitative perceptual study. This evaluation investigates listener perception of the types of variation captured by my learnt representation of $F_0$. I categorise what the perceived prosody relates to, such as: affect, information structure, or discourse structure. This knowledge can help determine what contextual information is needed when predicting appropriate prosody for each category of variation.

I present two approaches to learning structured representations of $F_0$, both of

which aim to capture both the categorical and continuous aspects of intonation. The proposed approach uses a variational autoencoder with a multi-modal prior. The second approach relies on more traditional clustering techniques and is used as a baseline. The baseline should indicate if the multi-modal prior learns a more interpretable or human-usable structure. In a discriminative subjective test I explore if different renditions from each system are *perceived* as distinct. And in the qualitative perceptual study I explore what information or intents the distinct renditions convey.

There is a gap between TTS research and prosody research. TTS research on controllability focuses on producing variation without deeper knowledge of how it is perceived. In contrast, more fundamental prosody research focuses on how acoustic-phonetic features map to linguistic categories and how this conveys meaning (e.g. via information structure) or relates to paralinguistic aspects of speech (e.g. speaker stance or emotion). This chapter helps bridge this gap by improving both controllability and our knowledge of what listeners perceive.

## 5.2 Related work

Controllable TTS has been approached from both supervised and unsupervised perspectives. Henter et al. (2018b) demonstrated that both can achieve the same quality for affect-related prosody. Unsupervised representation learning in TTS often uses continuous n-dimensional representations (Watts et al., 2015; Wan et al., 2019). However, continuous representations become increasingly difficult to interpret for $n \geq 3$. Poor interpretability limits the range of use cases. For example, Wang et al. (2018a) and Wan et al. (2019) are limited to transferring style from another natural utterance; human-in-the-loop control would be tedious.

Interpretability can be improved by keeping the number of dimensions small. Sun et al. (2020) use scheduled training to learn a 3-dimensional unsupervised representation that captures $F_0$, energy, and duration. An et al. (2021) demonstrate disentangled control of $F_0$, energy, and duration by minimising the mutual information of three independent 1-dimensional latents. These approaches produce representations not so different to existing acoustic correlates of prosody, which can be used to directly control a voice (Ribeiro and Clark, 2015; Wang, 2018; Klimkov et al., 2019; Mohan et al., 2021). Incorporating acoustic correlates

into representation learning models can lead to improved disentanglement and thus more interpretable representations.

Discrete prosody representations are another approach that can make control more usable. Ronanki et al. (2016a) propose using discrete intonation templates to control TTS voices. Each template is visually interpretable, making control user friendly for a human-in-the-loop. Tyagi et al. (2020) use the training data as prosodic templates and ranks them using a similarity metric; this provides a discrete form of control. Discrete representations are not necessarily interpretable; they may be abstract embeddings. If a representation is not interpretable, understanding what it captures requires an analysis of human perception. Unfortunately, when research shifts towards automated prosody prediction, interpretability is often not considered (Stanton et al., 2018; Karlapati et al., 2021).

Prosody should be modelled at the correct *domain*. While most approaches operate on sentences (Watts et al., 2015; Wang et al., 2018a; Henter et al., 2018b; Wan et al., 2019), the sentence domain may not be the most appropriate for a fixed-sized prosodic representation. Sentences contain a variable number of prosodic phrases. A sentence-domain representation will need to average over multiple prosodic constructions. Much less work has been conducted with prosodically-appropriate domains. Wang et al. (2019b) compare a discrete representation of $F_0$ at the phrase domain to shorter and longer domains. Reconstruction performance clearly shows that these fixed-sized representations are less accurate for longer domains, supporting the claim that longer domains contain more information on average.

Although claims of expressivity or prosody control are often made, variability or controllability are often not evaluated. Wan et al. (2019) use prosody reconstruction to measure the model's top-line performance, and prosody transfer is demonstrated qualitatively, but interpreting the latent space or choosing the best rendition were not tackled.

## 5.3 Discrete prosodic representation learning

To learn a discrete form of control in an unsupervised fashion it's necessary to enforce structure on the learnt representation. The approach taken here uses

variational inference with a prior that reflects the desired structure. I use a variational autoencoder (VAE) with a multi-modal prior. The learnt latent space of the VAE will be biased towards the prior's structure; it should be clustered across multiple components. Each component can be treated as a discrete intonation representation, these are referred to as '*intonation codes*'. To benchmark the proposed model against a more well studied clustering method, I introduce a baseline using an autoencoder and k-means clustering.

Before introducing the proposed VAE approach and the autoencoder baseline, we must consider the domain at which the representations are learnt.

### 5.3.1 Prosodic phrasing

In representation learning, temporal bottlenecks can be a powerful form of inductive bias that help guide what a representation captures. Wang et al. (2019b) demonstrated that changing the domain of a learnt representation—the temporal bottleneck—directly affects reconstruction performance. This is clear evidence that domain is an important design choice when using fixed-dimension representations. Learning a representation at a more prosodically-relevant domain may lead to improved interpretability.

In contrast to the previous chapters that use the utterance domain, this chapter uses the prosodic phrase domain. Unfortunately, accurately locating prosodic phrase boundaries (i.e. phrase breaks) requires manual annotation (Cole et al., 2017). While there is a correlation between syntactic and prosodic structure (Köhn et al., 2018), mismatches between syntactic and prosodic phrase boundaries are common (Ladd, 2008, Chapter 8). I use *chinks 'n chunks*, an heuristic parser designed for prosodic phrasing (Liberman and Church, 1992). This parser aims to identify contiguous units of text that map more closely to phrases for TTS than syntactic parsers do.

*Chinks 'n chunks* takes advantage of the right-branching nature of English: content words tend to occur towards the end of phrases and function words towards the beginning. However, since certain word types can behave like either, Liberman and Church (1992) define two categories:

| | | | | |
|---|---|---|---|---|
| **chink** | = | function words | + | tensed verbs |
| **chunk** | = | content words | + | objective pronouns |

Tensed verbs can behave like auxiliaries, thus starting a phrase. Objective pronouns can behave like nouns, thus acting as content words. The parsing algorithm is a simple greedy match: **{chink* chunk*}**. While this heuristic approach is not perfect, it provides a segmentation of utterances that is useful for imposing an inductive bias towards prosodic variation. Examples of utterances segmented into phrases are provided in Table 5.1.[1]

### 5.3.2 Probabilistic multi-modal latent space

To learn a prosodic representation, I extend the VAE model introduced in Chapter 3 by using a different prior distribution. A VAE's prior reflects our assumptions about the underlying latent factors that describe the data. The prior enforces structure on the latent space, making position, distance, and scale meaningful. Kingma and Welling (2013) use a uni-modal Gaussian prior, which makes the latent space smooth and allows for interpolation in that space (Berthelot et al., 2019).

The aim in this chapter is to uncover distinct prosodic behaviours in the data. This was inspired both by intonational phonology (Ladd, 2008; Ward, 2019) and the clustered intonation structure observed in Chapter 3 (Figure 3.8, pp. 84). Hence, I use a multi-modal prior to encourage a clustered structure: the variational mixture of posteriors (VAMP) prior (Tomczak and Welling, 2018). This is illustrated in Figure 5.1a in purple for 4 components. This prior makes the assumption that the data has multi-modal structure.

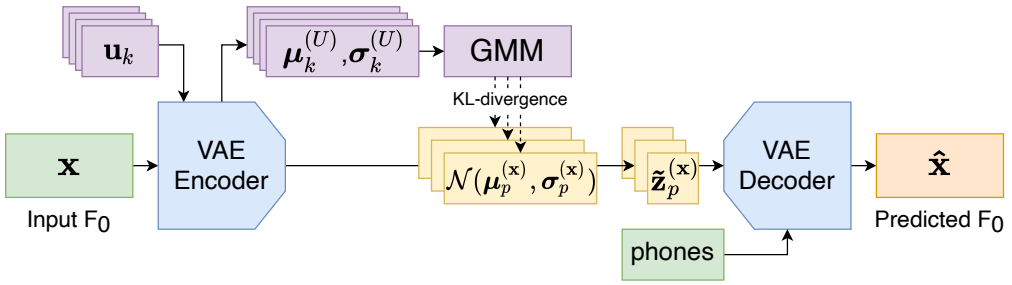
The VAMP prior is a Gaussian mixture model (GMM) whose parameters are learnt jointly with the rest of the model. It was derived as an approximation of the optimal prior: the aggregate posterior, which is a mixture of the training data's probabilistic embeddings. For the purposes of this study, the multi-modal structure is the necessary trait of the VAMP prior.

The name variational *mixture of posteriors* alludes to each component of the resulting GMM being defined by an approximate posterior, $q_{\boldsymbol{\phi}}(\mathbf{z}_k^{(U)} \mid \mathbf{u}_k)$. We do not directly learn GMM parameters, instead, to define the prior, we learn: the encoder $q$, parameterised by $\boldsymbol{\phi}$; and $K$ *pseudo-inputs* $U = \{\mathbf{u}_k\}_{k=1}^{K}$, where $K$ is

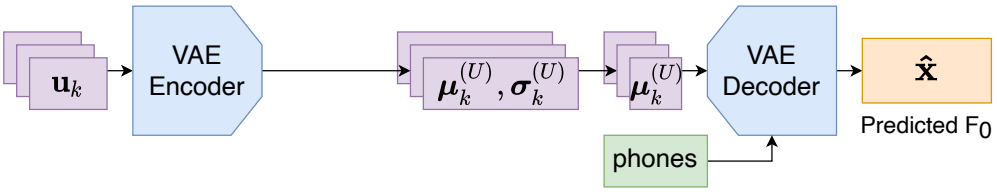
[1] To select the examples in Table 5.1, utterances were placed into three categories: utterances with 1 phrase, 2 phrases, and 3 or more phrases. For each test set book, 2 examples were randomly selected for each category.

Table 5.1: Utterances segmented into phrases using the *chinks 'n chunks* parser. Vertical bars represent phrase boundaries.

| Utterances with a single phrase |
|---|
| But Goldilocks wasn't good. |
| "I wonder who lives here?" |
| "What's the matter?" |
| "You think you're so clever?" |
| They tugged and tugged and tugged some more. |
| She hugged the farmer. |
| **Utterances with 2 phrases** |
| Goldilocks \| and the Three Bears. |
| Next, \| she tried the middle-sized chair. |
| "Nobody ever \| comes up here," moaned Sam. |
| Some people didn't \| believe him. |
| The farmer \| was very happy. |
| "I'll help!" \| he called. |
| **Utterances with 3 or more phrases** |
| There was a great, big father bear, \| a middle-sized mother bear, \| and a cuddly little baby bear. |
| "I don't believe \| you!" said Goldilocks, \| and put salt \| in the sugar pot. |
| "A wolf \| has come \| out of the forest!" |
| Puffing \| and panting, \| they reached the meadow. |
| This story \| has a farmer, \| his wife, \| their son Jack, \| a dog, \| a cat, \| a bird \| and an enormous turnip. |
| Slowly, slowly, very, very slowly, \| the turnip \| began to move. |

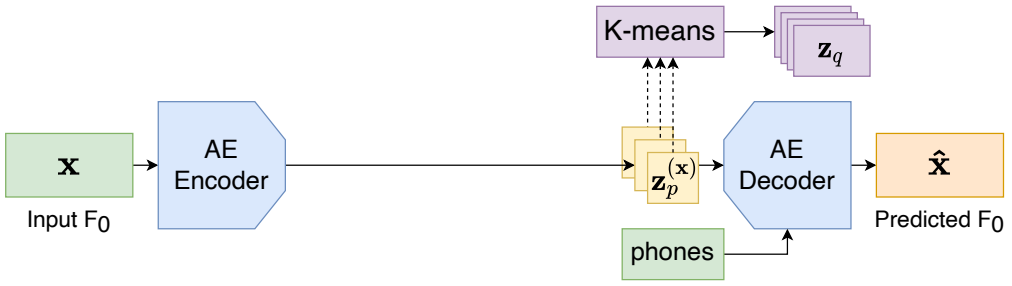



$\mathrm{VAE}_{\mathrm{VAMP}}$ training: $F_0$ is autoencoded through a variable length latent. The multi-modal prior is enforced using a KL-divergence with each phrase's approximate posterior.

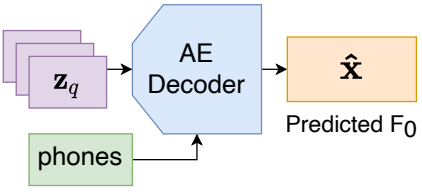



$\mathrm{VAE}_{\mathrm{VAMP}}$ synthesis: a *pseudo-input* must be chosen for each phrase. The mean of a GMM component is an *intonation code*.

(a) $\mathrm{VAE}_{\mathrm{VAMP}}$: probabilistic model with a learnt multi-modal prior.




$\mathrm{AE}_{\mathrm{K-MEANS}}$ training: $F_0$ is autoencoded through a variable length embedding. After training, clusters are computed using k-means.




$\mathrm{AE}_{\mathrm{K-MEANS}}$ synthesis: a cluster must be chosen for each phrase. A cluster centroid is an *intonation code*.

(b) $\mathrm{AE}_{\mathrm{K-MEANS}}$: baseline which first learns an embedding space and then extracts clusters.

Figure 5.1: Two systems used to learn *intonation codes*. Training and synthesis is illustrated for an utterance with 3 phrases, and a prior with 4 components.

a hyperparameter. The parameters $\boldsymbol{\lambda} = \boldsymbol{\phi} \cup U$ are learnt jointly with the VAE's decoder using the loss introduced in Equation 5.2. Importantly, the number of pseudo-inputs, $K$, is unrelated to the number of phrases in an utterance, $P$.

The pseudo-inputs, $\mathbf{u}_k$, are not real $F_0$ inputs, they are parameters learnt through backpropagation. However, because they are used as input to the encoder, they exist in the same vector space as the encoder's $F_0$ inputs. The pseudo-inputs are encoded, giving the variational parameters of the approximate posterior: $\boldsymbol{\mu}_k^{(U)}, \boldsymbol{\sigma}_k^{(U)}$. Each component of the GMM prior is defined using these means and variances. The mixture weights are fixed and uniform: $\pi_k = \frac{1}{K}$. Thus the VAMP prior is a mixture of the approximate posteriors of the learnt pseudo-inputs,

$$p_{\boldsymbol{\lambda}}(\mathbf{z}) = \frac{1}{K} \sum_{k=1}^{K} q_{\boldsymbol{\phi}}(\mathbf{z}_k^{(U)} \mid \mathbf{u}_k) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(\mathbf{z}_k^{(U)}; \boldsymbol{\mu}_k^{(U)}, \boldsymbol{\sigma}_k^{(U)}) \tag{5.1}$$

Tomczak and Welling (2018) demonstrated this new prior for fixed-size images. I present the first application of VAMP to variable sized sequence data: $F_0$ contours. This introduces a new challenge: learning a sequence of parameters for each pseudo-input. While it may be possible to model the sequence lengths as a random variable, I fix the number of frames in each pseudo-input at initialisation. In Section 5.4.1, I discuss pseudo-input sequence lengths and the challenges they pose in greater detail.

In this model, an utterance is represented using a sequence of latent variables, $\left\{\mathcal{N}(\mathbf{z}_p^{(\mathbf{x})}; \boldsymbol{\mu}_p^{(\mathbf{x})}, \boldsymbol{\sigma}_p^{(\mathbf{x})})\right\}_{p=1}^{P}$, one for each phrase in the utterance, where phrases are determined using the *chinks 'n chunks* parser. The variational parameters, $\boldsymbol{\mu}_p^{(\mathbf{x})}, \boldsymbol{\sigma}_p^{(\mathbf{x})}$, are the encoder outputs at the final time-step (i.e. frame) of a phrase. Unlike the approximate posterior parameters used in the prior, these parameters are derived from the $F_0$ contour, $\mathbf{x}$, hence the superscript: $\boldsymbol{\mu}_p^{(\mathbf{x})}$. The encoder models the latent variables for all phrases jointly: $q_{\boldsymbol{\phi}}(\{\mathbf{z}_p^{(\mathbf{x})}\}_{p=1}^{P} \mid \mathbf{x})$. The decoder reconstructs the $F_0$ contour, $\hat{\mathbf{x}}$, of an utterance using each phrase's latent variable jointly: $\log p_{\boldsymbol{\theta}}\left(\mathbf{x} \mid \{\tilde{\mathbf{z}}_p^{(\mathbf{x})}\}_{p=1}^{P}, \mathbf{c}\right)$, where $\tilde{\mathbf{z}}_p^{(\mathbf{x})} \sim \mathcal{N}(\mathbf{z}_p^{(\mathbf{x})}; \boldsymbol{\mu}_p^{(\mathbf{x})}, \boldsymbol{\sigma}_p^{(\mathbf{x})})$, and $\mathbf{c}$ is the phonetic conditioning features.

During training, each latent, $\mathbf{z}_p^{(\mathbf{x})}$, is indirectly compared with all $K$ *psuedo-inputs* through a direct comparison with the GMM components, i.e. the *psuedo-input's* approximate posterior, $q_{\boldsymbol{\phi}}(\mathbf{z}_k^{(U)} \mid \mathbf{u}_k)$. This direct comparison is the KL-divergence with the prior, $D_{KL}(\mathcal{N}(\mathbf{z}_p^{(\mathbf{x})}; \boldsymbol{\mu}_p^{(\mathbf{x})}, \boldsymbol{\sigma}_p^{(\mathbf{x})}) \,||\, p_{\boldsymbol{\lambda}}(\mathbf{z}))$, performed within the

evidence lower bound,

$$\log p_{\boldsymbol{\theta}}(x) \geq \mathbb{E}_{\{\tilde{\mathbf{z}}_p^{(\mathbf{x})} \sim \mathcal{N}(\mathbf{z}_p^{(\mathbf{x})}; \boldsymbol{\mu}_p^{(\mathbf{x})}, \boldsymbol{\sigma}_p^{(\mathbf{x})})\}_{p=1}^{P}} \left[\log p_{\boldsymbol{\theta}} \left(\mathbf{x} \mid \{\tilde{\mathbf{z}}_p^{(\mathbf{x})}\}_{p=1}^{P}, \mathbf{c}\right)\right] - \frac{1}{P} \sum_{p=1}^{P} D_{KL}(\mathcal{N}(\mathbf{z}_p^{(\mathbf{x})}; \boldsymbol{\mu}_p^{(\mathbf{x})}, \boldsymbol{\sigma}_p^{(\mathbf{x})}) \mid\mid p_{\boldsymbol{\lambda}}(\mathbf{z})) \tag{5.2}$$

This model is $\text{VAE}_{\text{VAMP}}$, and it has an autoencoder structure. $\text{VAE}_{\text{VAMP}}$ encodes and reconstructs mean-variance normalised log $F_0$, delta, and delta-delta features. Without a prosody predictor, some method of picking renditions, $\tilde{\mathbf{z}}_p$, from the prior must be devised. In Chapter 3, a sampling strategy was used to explore low probability renditions. However, the questions investigated in this chapter relate to discrete prosodic structure. Therefore, I treat each GMM component of the prior, $\mathcal{N}(\mathbf{z}_k^{(U)}; \boldsymbol{\mu}_k^{(U)}, \boldsymbol{\sigma}_k^{(U)})$, as representing a discrete prosodic behaviour. Sampling from a component, $\tilde{\mathbf{z}}_k^{(U)} \sim \mathcal{N}(\mathbf{z}_k^{(U)}; \boldsymbol{\mu}_k^{(U)}, \boldsymbol{\sigma}_k^{(U)})$, may produce variation within that rendition's style. However, to simplify evaluation, I do not evaluate renditions using random samples from a component. I only consider renditions using the component peaks: $\tilde{\mathbf{z}}_k^{(U)} = \boldsymbol{\mu}_k^{(U)}$.

The *intonation codes* for $\text{VAE}_{\text{VAMP}}$ are defined using the GMM component means: $\{\boldsymbol{\mu}_k^{(U)}\}_{k=1}^{K}$.

### 5.3.3 Baseline: two-stage clustering

Since the application of the VAMP prior is novel for both sequence and speech data, I propose $\text{AE}_{\text{K-MEANS}}$, a more traditional model that will serve as a baseline. $\text{AE}_{\text{K-MEANS}}$ uses an architecture made up of well studied components: an autoencoder and k-means clustering. The baseline has a similar model structure to $\text{VAE}_{\text{VAMP}}$, since the more details the models share, the more confident we can be that any performance differences are a result of the clustering approach.[2] $\text{AE}_{\text{K-MEANS}}$ has two training stages, connected by dashed arrows in Figure 5.1b.

[2]It is important to explain a significant difference between the two models: $\text{VAE}_{\text{VAMP}}$ uses variational inference, i.e. a probabilistic latent space, whereas $\text{AE}_{\text{K-MEANS}}$ does not. If a prior was added to $\text{AE}_{\text{K-MEANS}}$ it would impose structural assumptions and could effect the resulting clusters found by k-means. For $\text{VAE}_{\text{VAMP}}$, the prior directly enforces clusters. However, a prior can also obscure clusters, for example: a standard normal prior, $\mathcal{N}(0, 1)$, assumes there is only one cluster; and a uniform prior, $\mathcal{U}(0, 1)$, assumes everything is equally likely. Therefore, to ensure clustering is performed only by k-means, $\text{AE}_{\text{K-MEANS}}$ does not use a prior; $\text{AE}_{\text{K-MEANS}}$ uses an autoencoder, not a VAE.

1. An autoencoder learns to represent each utterance with a sequence of embeddings: $\{\mathbf{z}_p^{(\mathbf{x})}\}_{p=1}^{P}$, one embedding for each phrase in the utterance. Similar to $\text{VAE}_{\text{VAMP}}$, a phrase embedding, $\mathbf{z}_p^{(\mathbf{x})} \in \mathbb{R}^n$, is the encoder output at the phrase's final time-step (i.e. frame). Additionally, like $\text{VAE}_{\text{VAMP}}$, an utterance's pitch contour, $\mathbf{x}$, is reconstructed using each phrase's embedding jointly: $\hat{\mathbf{x}} = \mathit{decoder}(\{\mathbf{z}_p^{(\mathbf{x})}\}_{p=1}^{P}, \mathbf{c})$, where $\mathbf{c}$ is the phonetic conditioning features.

2. k-means clustering is used to cluster all phrase embeddings in the training data into $K$ clusters (Lloyd, 1982). This provides $K$ cluster centroids $\{\mathbf{z}_q\}_{q=1}^{K}$.

The *intonation codes* for $\text{AE}_{\text{K-MEANS}}$ are defined using the cluster centres, $\{\mathbf{z}_q\}_{q=1}^{K}$.

## 5.4 Experiments

Before introducing the evaluations in Section 5.4.2, I first discuss the technical details of the two systems, including challenges faced during model training.

### 5.4.1 System details

Models were implemented using Morgana (Hodari, 2020a) and the data was prepared using tts-data-tools (Hodari, 2020b).[3] Models are trained using the Usborne children's audiobook dataset. Details about the data can be found in Section 2.4.3.

$\text{VAE}_{\text{VAMP}}$'s architecture is show Figure 5.1a. Its encoder takes mean-variance normalised log $F_0$, delta, and delta-delta features as input. The encoder isn't conditioned on linguistic information as this would mean learning *pseudo-inputs* in linguistic input space as well as $F_0$ space. The encoder architecture uses a feedforward layer with 256 units, followed by three recurrent layers using gated recurrent cells with 64 units. These layers are clocked at the frame-level. To get the sequence of phrase-level approximate posteriors for an utterance, the embeddings at the last frame of each phrase are used (the remaining embeddings are decimated). Each embedding is projected to 32 dimensions, representing two 16-dimensional vectors: the mean and log variance of a 16-dimensional approximate posterior.

[3] Code and models are available at github.com/ZackHodari/discrete_intonation

The pseudo-inputs are defined in $F_0$ space and are passed through the same encoder described above, this provides the means and variances that make up the GMM prior. However, the encoder expects the pseudo-inputs to be sequences. Since pseudo-inputs are parameters learnt through backpropagation, we must define their shape when initialising the model. This means we must define a fixed sequence length for each pseudo-input. For each of the 20 pseudo-inputs, $\{\mathbf{u}_k\}_{k=1}^{20}$, the sequence length, $n_k = |\mathbf{u}_k|$, is defined as follows,

$$n_k = \begin{cases} 50k & \text{if } 1 \leq k \leq 10 \\ 50(k-10) & \text{if } 11 \leq k \leq 20 \end{cases} \tag{5.3}$$

That is, the first 10 pseudo-inputs in $\text{VAE}_{\text{VAMP}}$ have sequence lengths ranging from 50 to 500 frames, inclusive, with a step size of 50. The second 10 pseudo-inputs have the same sequence lengths as the first 10. Each sequence length was used twice to allow for multiple modes at each length. Determining the sequence lengths was a key challenge in training $\text{VAE}_{\text{VAMP}}$.

$\text{VAE}_{\text{VAMP}}$'s decoder takes two inputs: a latent sample from each approximate posterior (one for each phrase), and phone identity. Each phrase's latent sample is upsampled to phone domain. The phone-level latent sample and the phone identity are concatenated and upsampled to frame-level using ground-truth durations from forced alignment. I found that using a full linguistic specification (Table A.1, pp. 168) limited the range of $F_0$ variation captured by the discrete categories, hence the use of phone identity instead of linguistic feature vectors. The decoder architecture uses a feedforward layer with 256 units, followed by three recurrent layers using gated recurrent cells with 64 units, finally this is projected to 3 dimensions. The output represents mean-variance normalised log $F_0$, delta, and delta-delta predictions.

$\text{AE}_{\text{K–MEANS}}$ shares many architectural details with $\text{VAE}_{\text{VAMP}}$. $\text{AE}_{\text{K–MEANS}}$'s encoder and decoder in Figure 5.1b use the same architecture as $\text{VAE}_{\text{VAMP}}$'s encoder and decoder. The encoder outputs a 16-dimensional embedding (compared to 32 dimensions for $\text{VAE}_{\text{VAMP}}$) as it does not model uncertainty in the embedding space. The encoder's inputs are mean-variance normalised log $F_0$, delta, and delta-delta features, and the decoder reconstructs these three features. For consistency with $\text{VAE}_{\text{VAMP}}$, the encoder in $\text{AE}_{\text{K–MEANS}}$ does not use phonetic conditioning and the decoder is conditioned on phone identity. Conditioning the

decoder on a full linguistic specification limited the range of $F_0$ variation captured, as observed with $\text{VAE}_{\text{VAMP}}$.

I experimented with a variety of values for the number of pseudo-inputs in $\text{VAE}_{\text{VAMP}}$ and number of clusters in $\text{AE}_{\text{K–MEANS}}$. To make evaluation fair, I use the same number for each. I chose 20 pseudo-inputs and 20 clusters based on two factors: reconstruction performance, and range of variation visible in plots similar to Figure 5.2 (pp. 127).

The 20 intonation codes for $\text{VAE}_{\text{VAMP}}$ are the means of the pseudo-inputs' approximate posteriors: $\{\boldsymbol{\mu}_k^{(U)}\}_{k=1}^{20}$. The 20 intonation codes for $\text{AE}_{\text{K–MEANS}}$ are the cluster centroids: $\{\mathbf{z}_q\}_{q=1}^{20}$.

Both models are trained using $F_0$ features extracted from the waveforms with a 5 ms frame-shift using REAPER (Talkin, 2015). They were trained for 100 epochs using the Adam optimiser (Kingma and Ba, 2014) with a learning rate increasing linearly from 0.0 to 0.005 over the first 8 epochs and then decaying proportional to the inverse square of the number of batches (Vaswani et al., 2017, Sec 5.3). The batch size was 32. The KL-divergence term in $\text{VAE}_{\text{VAMP}}$ was weighted by zero during the first 5 epochs and increased linearly to 0.001 over 20 epochs. $\text{VAE}_{\text{VAMP}}$ converged to a KL-divergence of 5.32.

To perform synthesis, MLPG is used to generate an $F_0$ contour using the global standard deviation (Tokuda et al., 2000). This $F_0$ contour is synthesised into a waveform using natural durations and spectral features with the WORLD vocoder (Morise et al., 2016). At inference time, intonation codes must be selected, this can either be based on the reference $F_0$ contour, or through human-defined control.

To evaluate the the information captured by the intonation codes, I measured reconstruction performance on the validation set using the "oracle" intonation codes. The oracle intonation codes are derived using a reference $F_0$ contour by encoding the $F_0$ and assign each phrase's representation to the closest intonation code. The closest code is defined differently for $\text{VAE}_{\text{VAMP}}$ and $\text{AE}_{\text{K–MEANS}}$. For $\text{VAE}_{\text{VAMP}}$, we can compute the likelihood of a phrase's approximate posterior coming from each component of the prior. The oracle intonation code for $\text{VAE}_{\text{VAMP}}$ is the mean of the component with the highest likelihood. For $\text{AE}_{\text{K–MEANS}}$, the 1-nearest neighbour classifier can be used to assign each phrase

to the closest cluster. When reconstructing the validation set with oracle intonation codes, $\text{VAE}_{\text{VAMP}}$ and $\text{AE}_{\text{K–MEANS}}$ achieved an $F_0$ RMSE of 37.1 Hz and 33.0 Hz, respectively. Together, these objective results suggests that $\text{VAE}_{\text{VAMP}}$ successfully learns to reconstruct the inputs like $\text{AE}_{\text{K–MEANS}}$, but its performance is slightly reduced due to the structure enforced by the prior.

#### 5.4.1.1 Challenges training $\text{VAE}_{\text{VAMP}}$

The VAMP prior within the $\text{VAE}_{\text{VAMP}}$ model proved difficult to train. This is likely for two reasons. First, enforcing more structure on a latent variable increases the difficultly of encoding useful information, making posterior collapse more likely than in a standard VAE. Second, the VAMP prior has a new failure mode: the pseudo-inputs can collapse onto each other. With the VAMP prior, posterior collapse occurs in two stages: the pseudo-inputs converge towards each other, resulting in their approximate posteriors having the same means; and then typical posterior collapse occurs with a prior that is now effectively uni-modal.

To explore the behaviour of a model using the VAMP prior and to understand how to train a stable model, I assembled a toy dataset from LibriTTS (Zen et al., 2019). LibriTTS is an audiobook dataset derived from the LibriVox public domain audiobook library.[4] The toy dataset consisted of 481 utterances from 2 speakers, one male and one female, with distinct mean $F_0$.

The first test used a modified prior, called VAMP-data (Tomczak and Welling, 2018). This simply replaces pseudo-inputs with real inputs, meaning they are fixed and no longer learnable parameters—this reduces the difficulty of training. The data used can be prototypical examples of categories of interest, I used two pseudo-input data points: a random utterance from the male speaker, and a random utterance from the female speaker. In this simple scenario, other model hyperparameters could be tuned more easily and a model that did not suffer posterior collapse was successfully trained. The KL-divergence annealing schedule was particularly important in avoiding collapse. I developed the stable schedule using this toy data and the VAMP-data prior using fixed prototypical pseudo-inputs.

Following this, an initial model using learnable pseudo-inputs with a sequence length of 1 was investigated, i.e. using the VAMP prior, not the VAMP-data prior

[4] LibriVox data is accessible at librivox.org

with prototypical examples. However, this consistently resulted in posterior collapse despite much tuning. Visualising the latent space with principal component analysis (Pearson, 1901) clearly showed the pseudo-input's means converging together. Increasing the sequence length of the pseudo-inputs resulted in models that could train successfully. After moving back to the Usborne dataset, I was able to train a stable $\text{VAE}_{\text{VAMP}}$ model. I experimented with different pseudo input sequence lengths and found the more reliable $\text{VAE}_{\text{VAMP}}$ models used a sequence length that was similar to the phrase durations observed in the data. The duration of phrases in the data was between ~50 and ~500 frames.

Through more experimentation, I discovered that using different sequence lengths for some pseudo-inputs resulted in a larger amount of variation being captured. I observed this visually by creating plots similar to Figure 5.2. This finding led to the proposed $\text{VAE}_{\text{VAMP}}$'s sequence lengths, detailed in Equation 5.3.

#### 5.4.1.2 Joint duration modelling

The experiments that follow are performed using the $F_0$ model already described. However, versions of $\text{VAE}_{\text{VAMP}}$ and $\text{AE}_{\text{K-MEANS}}$ that jointly model $F_0$ and duration were also trained. Joint modelling means a single model predicted both $F_0$ and duration, and the parameters of this model were optimised according to the $F_0$ and duration losses jointly.

To account for the skewed distribution of phone durations, median durations were modelled using a transition distribution, following Henter et al. (2017b). The transition distribution describes the probability at each frame that a new phone begins at the next frame (i.e. do we transition to the next phone). Thus, the transition distribution models binary sequence data. At training time, natural durations were used to upsample phonetic inputs used by the decoder. $F_0$ and durations (specifically transition probabilities) are predicted using the same decoder and the reconstruction losses are optimised jointly. This model is similar to an S2S model, but replacing attention with an explicit duration model.

This model was trained successfully, and it was of an equivalent quality to the models presented in the previous section. However, this work was conducted after the evaluations described below. While evaluating the joint $F_0$ and duration models would surely produce interesting findings, designing and running evaluations involved a substantial amount of additional work. As such, it was not possible

to validate the joint $F_0$ and duration models. The experiments that follow use $VAE_{VAMP}$ and $AE_{K\text{–}MEANS}$ systems that only model $F_0$, and not duration.

### 5.4.2 Evaluation

The aim in this chapter is to capture distinct behaviours using *intonation codes* and to understand what these correspond to perceptually, such as: expressivity, dialogue structure, or information structure. To evaluate interpretability, the first step is to determine whether the codes produce perceivably distinct variation. Following this, we can explore what is captured by the codes that do produce distinct behaviours.

To synthesise using these models, intonation codes must be selected—one per prosodic phrase (*chink 'n chunk*-based). We can use the oracle intonation codes if a reference $F_0$ contour is available, or we could use human-in-the-loop control. However, the aim in this study is to systematically explore all renditions, not evaluate the most appropriate ones.

Unfortunately, for more than one phrase it is difficult to evaluate all sequences of codes as the number of combinations is exponential. In addition, we cannot pick sequences of codes randomly, as consecutive phrases could have conflicting prosodic behaviour. This issue does not stem from a limitation in the models—both of which are trained using multi-phrase utterances. Instead, the issue is the unknown grammar over a model's codes. A "language model" over the codes would be necessary to sample random code sequences without obvious conflicts between phrases. To simplify this systematic exploration of all codes, I restrict the evaluation stimuli to utterances with one phrase. This avoids any risk of choosing renditions that have conflicting prosodic behaviours across consecutive phrases.

The test set consists of 12 single-phrase utterances, chosen randomly from the 3 test set books (4 utterances from each): Goldilocks and the Three Bears, The Boy who Cried Wolf, The Enormous Turnip. The test utterances are presented in Figure 5.5 (pp. 131).

#### 5.4.2.1 Distinctiveness evaluation

To evaluate if the learnt representations capture different prosodic behaviour, listeners were presented with a pair of renditions of the same utterance from the

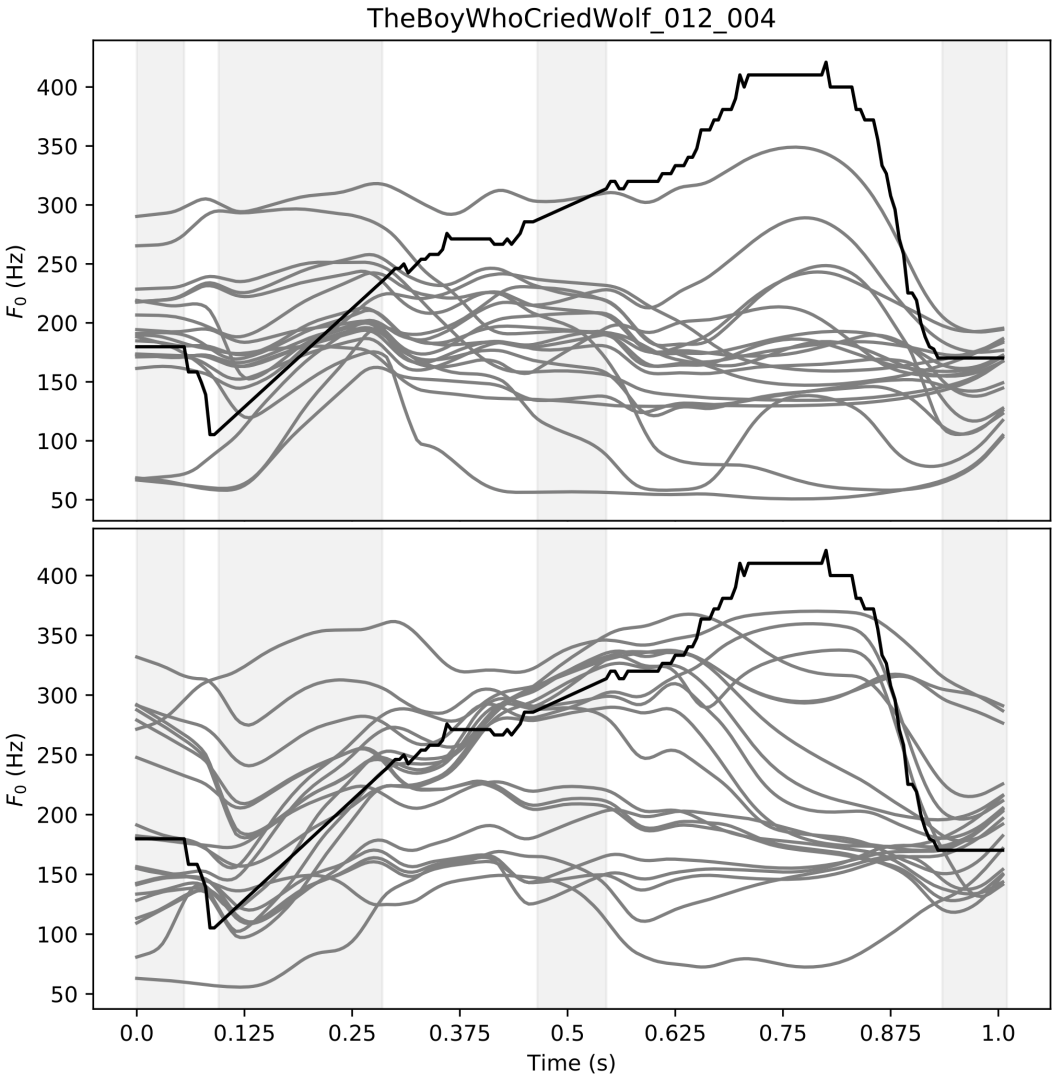



Figure 5.2: 20 *intonation codes* for $\mathrm{AE}_{\mathrm{K\text{–}MEANS}}$ (top) and $\mathrm{VAE}_{\mathrm{VAMP}}$ (bottom) for the utterance: "What's the matter now?". Unvoiced regions are shown by a slight shading along the x-axis. The black line shows natural $F_0$, for unvoiced regions the $F_0$ is interpolated linearly.

same system, and asked a forced choice question: "Decide if the two renditions have different intonation." For each of the 12 test utterances, 40 different renditions were synthesised using all intonation codes: 20 for $\mathrm{VAE}_{\mathrm{VAMP}}$, and 20 for $\mathrm{AE}_{\mathrm{K\text{–}MEANS}}$. Examples of the variation captured by these renditions can be seen qualitatively for one test set utterance in Figure 5.2. Evaluating distinctiveness for all 380 pairs of codes for each system is not feasible. Instead, 38 pairs of renditions were evaluation for each system: 38 pairs were selected randomly for $\mathrm{VAE}_{\mathrm{VAMP}}$, and a different 38 pairs were selected randomly for $\mathrm{AE}_{\mathrm{K\text{–}MEANS}}$.

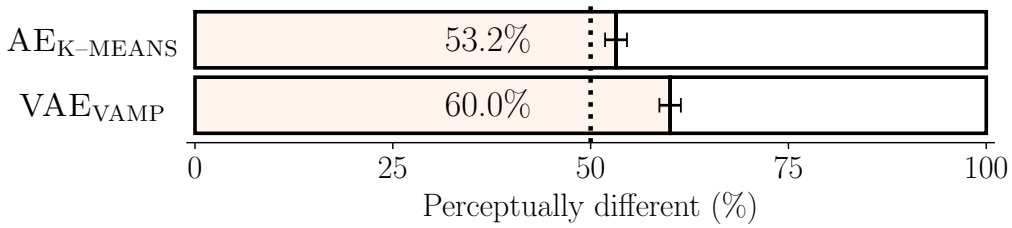



Figure 5.3: Same/different distinctiveness results for the two discrete prosody control systems. Listeners were presented with two renditions of the same single-phrase utterance for the same system and asked to "Decide if the two renditions have different intonation." Error bars show binomial confidence intervals.

A single screen for the listening test shows two renditions of an utterance from the same system. These the two renditions correspond to the two intonation codes from one of the 76 code pairs. A 2x2 Latin Square between-subjects design was used so that each listener heard all utterances, half the code pairs from $\text{VAE}_{\text{VAMP}}$, and half the code pairs from $\text{AE}_{\text{K–MEANS}}$. Across two listeners, all pairs were presented once. 22 native English-speaking participants each took around 45 minutes to complete the test, for which they were paid £8. This results in 11 "virtual listeners" that completed the full test according to the Latin Square design.

The distinctiveness results are presented in Figure 5.3. The error bars signify the confidence interval of a binomial significance test for each system. The rate of perceptual difference for $\text{VAE}_{\text{VAMP}}$ (60.0%) is significantly more than for $\text{AE}_{\text{K–MEANS}}$ (53.2%). To test individual intonation code pairs, I perform binomial significance tests for all 38 $\text{VAE}_{\text{VAMP}}$ code pairs and all 38 $\text{AE}_{\text{K–MEANS}}$ code pairs. This is followed by Holm-Bonferroni correction over all 76 p-values. After the correction, 16 pairs for $\text{VAE}_{\text{VAMP}}$ and 10 pairs for $\text{AE}_{\text{K–MEANS}}$ show significant perceptual difference (corrected $p < 0.005$). The full results of all 76 pairs can be found in Appendix B.

These results demonstrate that $\text{VAE}_{\text{VAMP}}$ learns more distinct representations, supporting the increased variation observed qualitatively in Figure 5.2. Given this, the interpretability experiment that follows exclusively explores $\text{VAE}_{\text{VAMP}}$.

#### 5.4.2.2 Interpretability evaluation

My aim is to understand what types of prosodic variation are captured by $\text{VAE}_{\text{VAMP}}$'s intonation codes. Since $\text{VAE}_{\text{VAMP}}$ learns representations in an unsupervised fashion there are no labels to compare against—an evaluation methodology exploited with emotion labels in Chapter 4. It may be possible to use taxonomies of prosodic forms, such as Goodhue et al.'s (2016) intonational bestiary, or prosodic constructions more generally (Ward, 2019). However, limiting evaluation to previously identified categories or constructions risks missing other important types of variation captured by the model. Similarly, a narrow focus on specific linguistic or affective phenomena increases difficulty for non-expert listeners, and potentially introduces bias.

Therefore, to explore the question of interpretability, I carried out a qualitative study. The experiment focused on unprompted free-form responses to the stimuli in one-to-one interviews. Following my analysis of the results, three discussion points were identified: **(i)** whether the prosodic differences captured discourse structural, information structural, or affective differences in meaning; **(ii)** whether intonation codes were interpreted in a consistent way across utterances; and **(iii)** what types of variation in prosodic meaning are salient to non-expert listeners.

One-on-one interviews discussing the renditions for various intonation code pairs were conducted with each participant. I led the interviews. An interview consisted of 6 screens, each presenting one intonation code pair for $\text{VAE}_{\text{VAMP}}$ on all 12 test sentences. That is, participants were presented with 24 stimuli on a single screen: the 12 test sentences when performed using one of the intonation codes were shown on the left, and the 12 test sentences when performed using the other intonation code were shown on the right. Participants were told that the first rendition of each sentence corresponds to one "condition", and the second rendition corresponds to another "condition". 5 native English-speaking participants took part in the 45 minute interviews, for which they were paid £8.

The 6 code pairs chosen for the 6 screens were the 6 pairs with the largest percentage of "different intonation" judgements from the distinctiveness test, i.e. the top 6 rows in Figure B.2 (pp. 173). The average rate of perceptual difference across listeners in the distinctiveness test for these 6 code pairs are summarised

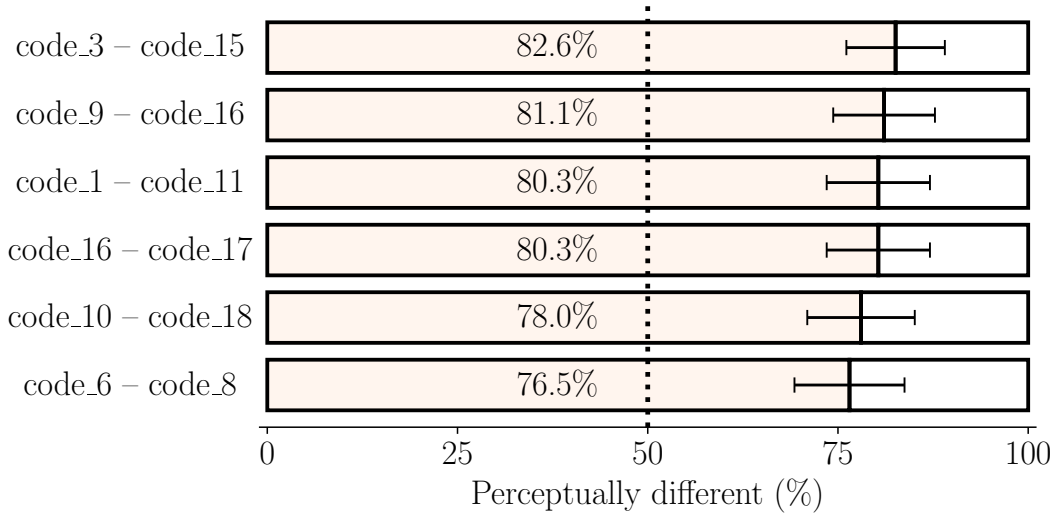



Figure 5.4: Distinctiveness results for top 6 intonation code pairs in $\mathrm{VAE}_{\mathrm{VAMP}}$. Error bars shows binomial confidence intervals.

in Figure 5.4. On average, each code pair is perceptually distinct in at least 9 out of the 12 test utterances.

During the first interview—a pilot test—it became clear that prompting the participant or answering their questions is highly likely to bias their responses. For example, if a participant asked "Should I comment on emotion?" answering "Yes." may lead to them focusing exclusively on emotion. The results presented only include content from the remaining 4 interviews. My instructions for these 4 interviews emphasised that the participant should comment on anything they wanted to about a stimuli. I also made it clear that they may choose to comment on how two renditions of an utterance differ, or simply on individual renditions in isolation. Participants were explicitly told they aren't required to comment on every rendition, only those for which they had something to say.

In the interviews, I took notes of the participants' comments. If a comment wasn't clearly understood, or if there was possible ambiguity, I would ask for a clarification, e.g. "Can you expand on that?" Alternatively, if comments were made on acoustic information—such as pitch changes—I would ask what, if any, effect this had or what meaning it added. Equally, if a participant commented on meaningful perceptual differences between renditions, I would ask if they are able to identify what they thought caused this difference.

| | |
|---|---|
| 13 | **There was no answer.** — statement, upset, surprised, anticipatory |
| 11 | **"I'm so hungry."** — upset, statement, continuation rise |
| 15 | **"Too hard!"** — question, statement |
| 10 | **They climbed the stairs.** — upset, continuation rise, anticipatory, sad, narrative |
| 20 | **"What's the matter now?"** — statement, question, rhetorical, annoyed, friendly, urgent |
| 11 | **"We'd better make sure."** — upset, question, "standard" style, uncertain |
| 12 | **"Do you think we're so stupid?"** — insulted, upset, rhetorical, sad |
| 19 | **"I'm sorry."** — fake apology, passive aggressive, question, apology, "standard" style, upset |
| 9 | **He wanted a turnip.** — statement, narrative, continuation rise, sad, bored |
| 7 | **They both tugged and tugged.** — narrative, upset, child storytelling, "standard" style |
| 11 | **But the turnip didn't move.** — upset, statement, narrative, surprised |
| 14 | **"It's enormous!" cried Jack.** — surprised, exclamation, childlike |

Figure 5.5: Descriptive terms used by participants for each test utterance. The terms listed include only those that were used multiple times for a given utterance. The column on the left indicates the total number of unique terms that were used to describe that utterance.

### (i) Prosodic behaviour captured

The interview transcriptions were summarised into descriptive terms based on keywords used by participants. 68 descriptive terms were found, with 26 terms being used to describe more than one utterance. The number of unique descriptive terms for each utterance and a list of more common terms used can be found in Figure 5.5. The full results are presented in Appendix B. The terms used to describe 4 or more utterances, in descending order of frequency, were: upset, statement, narrative, question, surprised, "standard" style, continuation rise,[5] emotional, anticipatory, sad, child storytelling, monotonous, and confused. The broad range of terms used demonstrates the variety of prosodic behaviour captured by $\text{VAE}_{\text{VAMP}}$'s intonation codes.

Most of the terms relate to more affect-related changes, this is not so surprising when considering the style of the Usborne data: children's audiobooks. Changes in interpretation relating to discourse structure and information structure were reported, most commonly, continuation rise. Stance and interaction

[5]The term "continuation rise" was not used directly. Participants described the effect typically as "holding the floor" in combination with a comment on pitch.

related effects were also present, e.g. back-channelling, insincere apology, feigning being impressed, feigning surprise, humorous sarcasm, and typical sarcasm.

#### (ii) Intonation code consistency

Certain intonation codes were consistently reported to produce styles such as: questioning, upset, and narrative. However, these were likely the most common styles in the data, meaning that codes were not wholly consistent; a code's interpretation often changed depending on the utterance. From Figure 5.5, we can infer that the codes are not entirely consistent, since certain utterances elicit terms that others do not.

The least descriptive utterances, such as "What's the matter now?" and "I'm sorry", elicited the most comments from participants. This correlation between number of unique terms and semantic ambiguity suggests that, unsurprisingly, semantics has a large impact on the perceived effect of the codes. This is either because the intonation codes are able to produce variation more freely, or because participants can imagine more contexts for ambiguous utterances.

To determine if individual codes behave consistently, a larger number of participants and utterances would be needed. Ideally, utterances should be specifically designed for the test to control for other potential perceptual effects.

#### (iii) Listener behaviour

In general, participants' interpretations appeared to be dependent on what contexts they thought were appropriate for a specific rendition. Some participants even provided rich descriptions of contexts a rendition might make sense in. This could be a useful direction for analysing what effects a representation captures. Instead of comparing distinct renditions, participants could be asked to describe a context or situation that a rendition might make sense in—selecting "unsure" or "invalid" when necessary. From this descriptive task we could categorise interpretations of different renditions and determine if renditions consistently correspond to plausible, and potentially uncommon, contexts.

Interestingly, participants perceived some timing and loudness changes, despite $F_0$ being the only feature that was modified. This is possible as timing and loudness are perceptual phenomena and a perceived change in these can occur

even if their acoustic correlates—duration and intensity—are not modified. This provides evidence that $F_0$ leads to perceptual changes in prosody that are broader than just intonation, empirically supporting the difference between acoustic correlates and perceptual correlates of prosody.

In some cases, participants described a behaviour, but noted it as inappropriate, most commonly for questions and continuation rises. This is to be expected since the codes were chosen arbitrarily, i.e. the utterance's content was not considered. For continuation rises, participants sometimes felt it was jarring to hear the voice hold the floor but to hear no additional speech following the continuation rise.

## 5.5 Conclusion

In this chapter, I introduced a new method for learning discrete representations of $F_0$ contours. I demonstrated my approach is more effective at learning distinct prosodic behaviours than traditional methods. The proposed system is a novel application of a multi-modal prior in a VAE, being the first use of this prior on sequence data. In addition, the model learns representations at a prosodically-relevant domain, this inductive bias should encourage the representations to capture more interpretable prosodic variation.

In a set of qualitative interviews, I investigated the interpretability of the learnt representations (**Theme 2**). The interpretation of different renditions varied based on semantics, where ambiguity led to users describing potential contexts based on what they perceived. A broad range of affective, and some information structural, variation was observed. Having understood what kinds of prosodic behaviours are captured by the representation, specific context information could be collected for use in prosody prediction. For example: expressive behaviours may be influenced by the connotation or meaning of a word; affective prosody may relate to interpersonal information; prosody related to information structure may benefit from parse tree information; and dialogue structure effects may require context information about turns, personality traits, and setting.

The qualitative interview structure used to investigate listener perception and interpretability of learnt representations is a useful direction for further research. By expanding this evaluation paradigm in a number of ways, it could be

used to validate unsupervised representations as an interpretable form of human-in-the-loop control. As discussed, using more participants would improve the ability to make statistical inferences, carefully designing the stimuli would allow for control of confounding linguistic variation, and verifying that the representations are consistent across stimuli would ensure they can be controlled efficiently by a human-in-the-loop. Nonetheless, the findings in the evaluations presented here are useful for determining the most relevant context information when considering how to automatically predict prosody. In the following chapter, I introduce additional context information to automatically predict appropriate prosodic representations (**Theme 3**).

# Chapter 6

# Prosody modelling using suprasegmental context

*This chapter covers the work in "CAMP: A two-stage approach to modelling prosody in context" (Hodari et al., 2021) presented at ICASSP 2021, Toronto, Canada.*

*The work in this chapter was performed during an internship at Amazon TTS Research, Cambridge, UK. The work presented here was completed entirely by me, with advice and discussion from all co-authors. I relied upon existing code, but contributed a significant amount of implementation to complete this work. Notable exceptions that were contributed by colleagues are: the compound noun processing, the idea for BiLSTM smoothing after upsampling, and the percentile bootstrap implementation.*

*Following the previous investigations of how to avoid current prosody modelling issues through controllability (**Theme 1**), I now explore the use of these methods in a TTS system without the use of human-in-the-loop control. This chapter works towards **Theme 3**: producing prosody that is appropriate to the context. I use syntactic and semantic context features to drive prosody prediction. The proposed framework is designed to make it easy to incorporate more context through parallel context encoders in a prosody predictor. Subjective results demonstrate that context can improve prosody quality very significantly, but that more context should be introduced to further improve prosody in TTS.*

*As part of this work, I learn a disentangled representation of prosody from the mel-spectrogram at the word domain. This is the first work in the literature to learn a representation from the spectrogram at a prosodically-relevant domain. I also investigated duration modelling in sequence-to-sequence (S2S) models. A preference test provided the first statistically signifiant result in the literature showing that duration modelling produces better prosody than attention.*

## 6.1 Introduction

In this thesis, I approach prosody modelling as two separate tasks: controllability and appropriateness. Compared to typical TTS systems, where prosody is modelled jointly with other acoustic information, the challenges of prosody modelling are more addressable within this framework. Prosody control is challenging because prosody is embedded in the acoustic signal alongside other information, such as segmental and channel information (Ladd, 2008). Predicting appropriate prosody is challenging because the model lacks prosodic context. When additional context *is* introduced, it is typically used inefficiently, such as to predict frame-level acoustic detail, instead of suprasegmental prosodic variation. To address the challenges of both tasks, I propose a context-aware model of prosody (CAMP). CAMP is trained in two stages; *stage-1* focuses on the entanglement of prosody, and *stage-2* focuses on the lack of context.

Suprasegmental prosody operates over different domains than other information in the acoustic signal. In particular, prosody varies more slowly than frame-level and segmental information, i.e. prosody operates over longer domains (Ward, 2019). To account for this, I learn a representation from the mel-spectrogram using a temporal bottleneck in *stage-1*. A prosodically-motivated domain for the temporal bottleneck leads to disentanglement of prosody from other information in the mel-spectrogram. The disentangled representation is used to drive a controllable TTS model, which can therefore produce multiple prosodic renditions, thus satisfying a condition established in the thesis: prosody must be controlled or it will be ignored.

Without sufficient context, predicting appropriate prosody is an ill-posed problem (Clark et al., 2019), as any number of prosodic renditions could be deemed appropriate for the sentence. Under the SPSS paradigm, models used lin-

guistic features consisting mostly of segmental information, but also some limited structural information at the syllable, word, and phrase domains (cf. Table A.1, pp. 168). While these linguistic features provide some context, it is insufficient for modelling prosody in expressive speech. Recent models have even less context, using only phone identity (Ren et al., 2020; Elias et al., 2021). As explored in Chapter 5, understanding what types of variation exists in the data could inform which specific context features are most relevant.

Suprasegmental prosody is influenced by a broad range of context information, from syntax and semantics to affect, pragmatics, and setting (Goodhue et al., 2016; Köhn et al., 2018). To improve appropriateness, we need suprasegmental context: information from domains that are above the phone domain, and ideally from surrounding phrases or utterances. Therefore, in *stage-2*, I propose a model that can incorporate arbitrary context information. This model, the "prosody predictor", uses a prosodically-relevant loss to ensure the context information is used to directly improve prosody prediction.

Many attempts to incorporate context use the additional context as input to an acoustic model with a frame-level spectrogram loss (Hayashi et al., 2019; Fang et al., 2019). This leads to inefficient use of the context compared to my proposed prosody predictor. Since current acoustic models do not perfectly minimise the spectrogram loss, during training the model will continue to focus on improving prediction of detailed acoustic content in each frame. Suprasegmental prosody varies over longer domains, meaning it has a weaker impact on the spectrogram loss. Therefore, prosody will be deprioritised and the additional context will be underutilised. Instead, the loss must be designed to add inductive bias that guides the model to what is most important: in this case, prosody. This could be achieved by: predicting acoustic correlates of prosody like $F_0$ (Wang et al., 2019b) (as explored in Chapters 3, 4, and 5), selecting prosodic templates (Tyagi et al., 2020), predicting a disentangled prosody representation (as explored here), or using self-supervised learning or contrastive learning losses that directly impose inductive biases towards longer-term variation (Baevski et al., 2020).

The representation learnt in *stage-1* was the first prosody representation in the literature to be learnt from the spectrogram at a prosodically-motivated domain. That is, my representation is not limited to a set of acoustic correlates of prosody, and it is not learnt at the phone or utterance domains—which would lead

to information content/capacity issues discussed below. In addition, the prosody predictor trained in *stage-2* was the first to use *additional* context to directly *predict* a prosody representation learnt from the spectrogram. I also present two sequence-to-sequence (S2S) baselines, using either attention or explicit durations in Section 6.4. I published the first results demonstrating a significant improvement attributable to the use of a duration model, compared to attention; this was later corroborated by Shen et al. (2020).

## 6.2 Related Work

Unsupervised representations of speech can be learnt from the spectrogram or waveform (van den Oord et al., 2017; Schneider et al., 2019; Dunbar et al., 2019). While there are many methods that use or define prosodic correlates (Suni et al., 2015; Klimkov et al., 2019; Ribeiro and Clark, 2015), unfortunately, there is less work on unsupervised representation learning specifically for prosody.

Prosody representations are typically learnt at the sentence domain (Wang et al., 2018a; Tyagi et al., 2020; Karlapati et al., 2021). Representations at the sentence domain are too coarse and cannot perfectly reconstruct prosody. To accurately capture prosody we need a sequence of representations, e.g. at the syllable domain (Wang et al., 2019b) or phrase domain (Chapter 5).

The linguistic linker introduced by Wang et al. (2019b) also performs the same prosody prediction task explored in this chapter, but with context limited to traditional linguistic features. Wang et al. (2019b) experiment with different domains when modelling $F_0$, clearly showing the importance of domain in representation learning. However, they do not consider other acoustic correlates of prosody, such as duration and intensity. Representations extracted from a spectrogram, as explored in this chapter, can capture these aspects of prosody.

Other works propose the use of a spectrogram-based representation, but at the sentence domain. Stanton et al.'s (2018) text-predicted global style tokens (TP-GST) uses an autoencoder structure to learn a representation from the mel-spectrogram. At inference time TP-GST predicts the representation using context. However, this context is limited to segmental information. Tyagi et al. (2020) learn representations in a similar way, using a reference encoder, and at inference time they use the training data as prosody templates and select tem-

plates using syntactic and acoustic context. Tyagi et al.'s (2020) is the first work with S2S models to use *additional* context information *directly* for prosody modelling. Learning to predict representations, like in TP-GST, may lead to better generalisation compared to selecting templates. However, it is important that any prosody modelling uses additional context, like in Tyagi et al.'s (2020) approach.

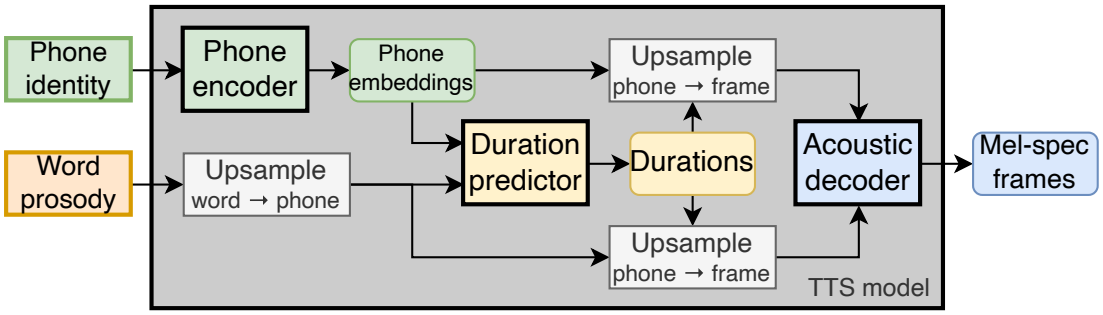
## 6.3 Two-stage prosody modelling

My proposed context-aware model of prosody, CAMP, shown in Figure 6.1c, uses context features to predict prosody representations. These predicted prosody representations drive a *"TTS model"*, detailed in Figure 6.1a. The *TTS model* is learnt during *stage-1* of training as part of an autoencoder. This audoencoder model, shown in Figure 6.1b, is the top-line system: ORA. ORA uses the oracle prosody representations, extracted using a word-level reference encoder, to drive the *TTS model*. In *stage-2* of training, the prosody predictor, used by CAMP, is learnt. The prosody predictor is trained to predict the oracle prosody representations using suprasegmental text-derived context features for the current sentence.

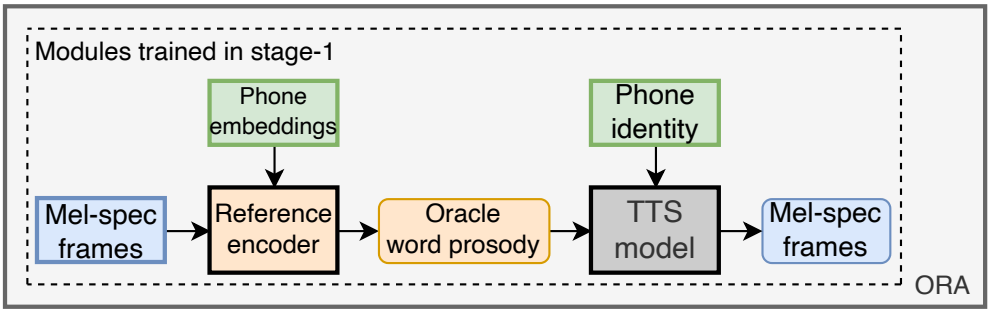
The core idea behind CAMP is that context information must be used to predict prosody—which exists over suprasegmental domains—as opposed to predicting lower-level acoustic information, such as the mel-spectrogram. Any prosody predictor must be designed with some inductive bias that focuses the model on the prosodic domain. In this chapter, I use a loss that explicitly focuses on prosody by predicting a word-level prosody representation. Training the model in two separate stages means that a disentangled representation of prosody is learnt and ensures that context information is used to predict prosody, not frame-level acoustic detail.

### 6.3.1 *Stage-1*: Word-level prosodic representation learning

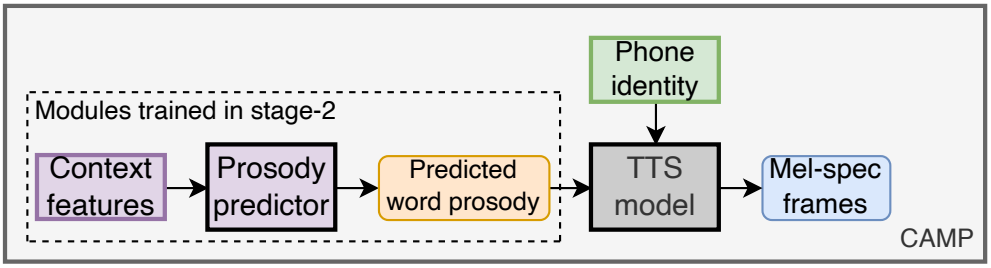
The prosody representation is learnt by ORA (Figure 6.1b), which has an autoencoder architecture made up of two components: the reference encoder, and the *TTS model*. The reference encoder architecture is shown in Figure 6.2a. The *TTS model* (Figure 6.1a) consists of multiple modules: a phone encoder (Figure 6.2c), duration predictor (Figure 6.2d), and acoustic decoder (Figure 6.2b).




(a) The *TTS model* synthesises speech according to the word-level prosody representations provided. This is the decoder of ORA, and is trained during *stage-1*.




(b) ORA is an autoencoder, using oracle prosody extracted from the reference mel-spectrogram. The encoder outputs a disentangled word-level representation of prosody. The decoder is a controllable *TTS model*. The dashed box illustrates *stage-1* of training: an autoencoding task with an information bottleneck. To make the diagram more readable, the phone embeddings are shown as an input to the reference encoder, in reality these are derived from the phone identity, as shown above in Figure 6.1a.




(c) CAMP uses prosody predicted by suprasegmental context. The dashed box illustrates *stage-2* of training: the learnt prosody representations are predicted using prosodic context.

Figure 6.1: Three speech synthesis systems. (a) *TTS model* driven by a prosody specification: the learnt word-level prosody representations. (b) ORA: top-line system using reference speech. (c) CAMP: the proposed system using context to predict the word-level prosody representations. Each subfigure shows the model's configuration for synthesis. The bashed boxes in (b) and (c) represent the two stages of training introduced in Section 6.3, the contents of these boxes illustrate which modules are trained in that stage.

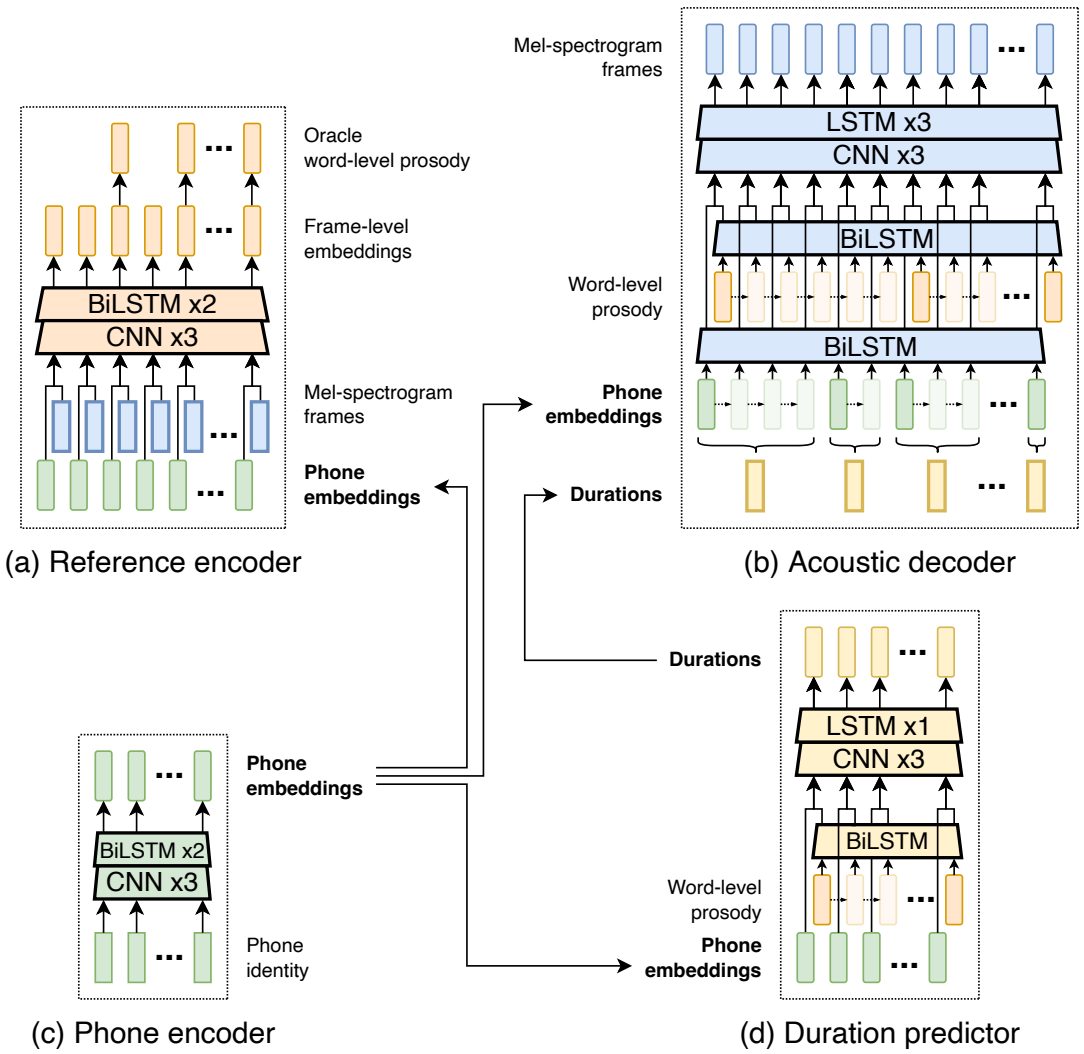



Figure 6.2: Details of modules used in Figure 6.1. The inputs and outputs for each module, along with differences in sequence lengths are illustrated. The connections between modules are shown for phone embeddings and durations. Modules are also connected by the word-level prosody representations, but this is not illustrated since the connection changes between ORA and CAMP. (a) The *reference encoder* learns a disentangled representation of prosody at the word level. (b) The *acoustic decoder* predicts the mel-spectrogram using either the oracle of predicted prosody representations. (c) The *phone encoder* learns a shared phonetic representation used by all other modules. (d) The *duration predictor* predicts phone and silence durations using either the oracle or predicted prosody representations.

During *stage-1*, ORA is trained. This involves autoencoding mel-spectrograms while disentangling prosody in the reference encoder's word-level output. ORA attempts to copy the oracle prosody using this reference encoder. Two design choices are made in ORA's reference encoder (Figure 6.2a) to encourage disentanglement: a **temporal bottleneck**, and **phonetic conditioning**. Note that after *stage-1* of training, the *TTS model* is frozen and used in the proposed system: CAMP.

A **temporal bottleneck** in the reference encoder is achieved by decimating the frame-level output, i.e. by taking the final embedding at the given domain. Determining the domain for the bottleneck involves a trade-off. Too short and segmental and background information will remain entangled. This is evident in van den Oord et al.'s (2017) experiments with unsupervised phonetic discovery where a high bit rate representation captures segmental information very effectively. Conversely, too long a domain will sacrifice descriptive power. This can be observed with Wang et al.'s (2018a) and Wan et al.'s (2019) reconstruction experiments where prosody is not faithfully reproduced for the sentence domain. Wang et al. (2019b) demonstrate clearly that the bottleneck's domain directly impacts reconstruction accuracy. As discussed in Chapter 5, sentences contain an arbitrary amount of information and are not a suitable domain for prosody. Ideally, the syllable or prosodic phrase would be used. However, for simplicity I use words as the temporal bottleneck's domain. Specifically, each word and each pause is represented with one fixed-length vector. In English the average speaker has a maximum speaking rate of 1.46 syllables per word (Flipsen Jr, 2006), thus using the word domain should provide a close approximation of a prosodic domain.

**Phonetic conditioning** simply means providing the reference encoder with phonetic information to aid disentanglement. The phonetic information is the same information provided to the acoustic decoder. Conditioning is complimentary to the temporal bottleneck, as it allows the reference encoder to avoid representing the now redundant phonetic information. For conditioning to benefit disentanglement the representation must have an information bottleneck. The prosody representation in the reference encoder is constrained in number of dimensions, as well as having a temporal bottleneck.

The *TTS model* uses the same phone encoder (Figure 6.2c) as Tacotron-2

(Shen et al., 2018). Instead of attention, I use an explicit duration model, similar to DurIAN (Yu et al., 2019) or FastSpeech-2 (Ren et al., 2020). Not only did using explicit durations remove disfluencies and other attention-related error modes, it also led to improved prosody.

The duration predictor (Figure 6.2d) takes both phone embeddings and the word-level prosody representations as input. This allows the duration predictor to generate durations according to the prosody specification. Considering the duration predictor is trained jointly with the reference encoder in *stage-1*, this design choice will ensure the reference encoder captures duration-related aspects of prosody in the learnt representation. At training time, durations are predicted, but they are only used to calculate the loss and gradients. Natural durations are used to upsample the word-level prosody representations to frame-level for the acoustic decoder; this is necessary in order to compute the acoustic loss. At synthesis time, predicted duration are always used to upsample the word-level representations. The upsampled prosody representations provided to the acoustic decoder contain repetitions, it was found that adding a BiLSTM (Schuster and Paliwal, 1997) to smooth this upsampled sequence improved performance.

The acoustic decoder (Figure 6.2b) follows a similar architecture to CopyCat's decoder (Karlapati et al., 2020). This doesn't use any autoregressive feedback of predictions, typically present in most S2S models, as the conditioning on a sequence of prosody representations was found to provide enough local context. Similar to the duration predictor, it was found that adding a BiLSTM to smooth the upsampled inputs improved performance. Thus, my acoustic decoder adds two BiLSTMs, one after the upsampled phone embeddings and one after the upsampled prosody representations.

### 6.3.2 *Stage-2*: Context-aware prosody prediction

In order to use the *TTS model* for synthesis, the oracle prosody representations cannot be used as these are derived from the reference audio. In *stage-2* of training, a prosody predictor is learnt in order to replace the reference encoder. The prosody predictor uses text-derived context features as input. The proposed system drives the *TTS model* using prosody predicted with context features: it is a context-aware model of prosody (CAMP).

This task of mimicking reference embeddings has been referred to as "linguistic linking" (Wang et al., 2019b), or "text-prediction" (Stanton et al., 2018). I build upon this idea, emphasising the need for suprasegmental features to see improvement on this task.

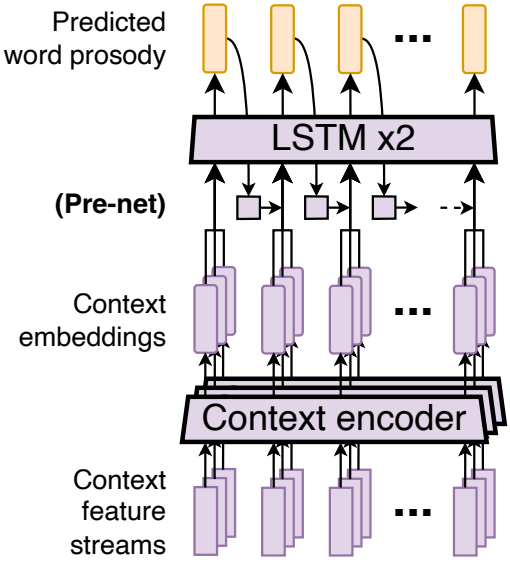
#### 6.3.2.1 Prosody predictor

The prosody predictor (Figure 6.3a) autoregressively predicts word-level prosody representations using information extracted from context features. By predicting a prosody representation the context should be used to improve prosody prediction, not to improve detail in the predicted spectrogram. Autoregression allows the prosody predictor to model how prosody transitions from one word to the next, like a language model over prosodic patterns. I focus on adding more features, as opposed to increasing context width by training on longer extracts. The prosody predictor uses one or more context encoders to incorporate different feature streams. I propose five context streams: four syntactic context features, and a semantic context encoder based on a pre-trained language model.

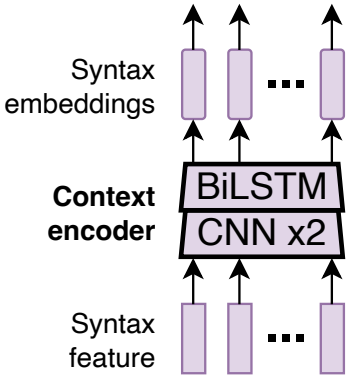
This work was in part inspired by van den Oord et al.'s (2017) experiments using a learnt prior with VQ-VAE. By using a high bit rate representation, their autoencoder learns low-level acoustic information and the learnt prior is able to capture phone classes in an unsupervised fashion. My prosody predictor can be interpreted as a learnt conditional prior, where the latent space—the prosody representation—has unknown variance. By using coarser-grained (i.e. word domain) representations my autoencoder, ORA, captures prosodic information, and the learnt prior, the prosody predictor, may capture prosodic patterns.

#### 6.3.2.2 Syntactic information

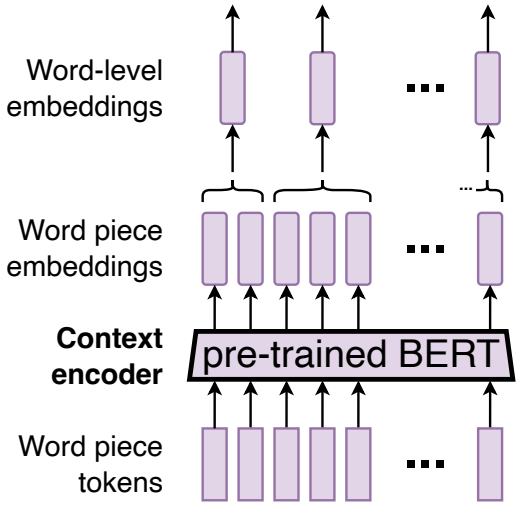
There is a correlation between syntax and prosody (Köhn et al., 2018). For example, from a perceptual point of view, prosody can disambiguate syntactic ambiguities (Allbritton et al., 1996). To take advantage of this relationship, I experiment with four syntax-related context features: part of speech, word class, compound noun structure, and punctuation structure. Each of these features were input to a separate *context encoder*. The architecture of a single syntactic context encoder is illustrated in Figure 6.3b.




(a) Prosody predictor.




(b) Syntax context encoder.




(c) Semantic context encoder.

Figure 6.3: Autoregressive prosody predictor and two context encoders. (a) The prosody predictor uses any number of context encoders to autoregressively predict the word-level prosody representations. All context embeddings must be at the word domain in this architecture. (b) Syntax context encoder with 2 CNN layers and a BiLSTM. The 4 syntax features each use a separate context encoder with this architecture. (c) Semantic context encoder using a pre-trained language model, BERT, to create the context embeddings. BERT uses WordPiece tokenisation, so the subword-level outputs are averaged to the word domain.

Part of speech (POS) represents the syntactic role of a word. While this does not capture all syntactic relationships present in a parse tree, it is easier to incorporate into a model than graph-structured inputs.[1] POS tags were extracted using the LAPOS tagger (Tsuruoka et al., 2011) for the Penn-Treebank tagset (Marcus et al., 1993). An additional tag for punctuation or silence was included to ensure the length of the tag sequence matched the number of words.

Word class is a coarser classification of POS: open class words (content words) and closed class words (function words). Compound noun structure was represented using a binary flag indicating if a word is part of a compound noun. Both word class and compound noun structure were extracted from the POS sequence, meaning they are redundant. Finally, punctuation structure simply represents if a word-level token is a punctuation mark. These binary features may prove useful for determining prosodic effects such as: emphasis, which is often placed on content words; stress, which changes in compound nouns; and pausing, which is often determined by punctuation.

#### 6.3.2.3 Semantic information

Semantics can also influence prosody. Depending on what is appropriate for the semantic context, prosody may be used to signal focus or givenness (Krifka, 2008), resolve ambiguities surrounding relative salience (Lewis, 1979), or convey connotations or attitudes relating to the theme and rheme of an utterance (Halliday and Matthiessen, 1999). Instead of relying on manually extracted features like the syntax features above, I utilise contextualised word embeddings from BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), as these correlate with semantic information (Rogers et al., 2020). BERT is part of a class of self-supervised language models. These models use self-supervised losses, such as masked prediction, to learn "contextualised embeddings" from the data without a specific task in mind. These embeddings are typically applied to downstream tasks, as explored here for prosody modelling.

BERT takes WordPiece tokens as input and uses self-attention layers to

[1] A collaboration conducted with Karlapati et al. (2021), in parallel to the research in this chapter, focused on the use of constituency parse trees as input to a graph neural network context encoder—we (Karlapati et al., 2021) used a similar prosody prediction framework. Similar to my results in Section 6.5.3.2, we found that syntactic information added no information to a BERT-based context encoder.

predict tokens that have been masked out, a task known as masked language modelling (Devlin et al., 2019). WordPieces are segments of words (Wu et al., 2016a), i.e. subword units. The more common a word or sequence of characters, the more likely it will be part of the WordPiece vocabulary. In this way, WordPieces can accommodate any number of words, including unknown words, using a fixed sized vocabulary. They can also handle certain morphological variation. Importantly, WordPieces mitigate the increased difficulty of language modelling on character-based vocabularies.

I use a pre-trained $\text{BERT}_{\text{BASE}}$ model as a context encoder (Figure 6.3c). The prosody predictor's autoregressive LSTM expects the sequence of semantic context embeddings to be at the word domain. However, $\text{BERT}_{\text{BASE}}$ outputs one embedding per WordPiece. To align this output with the word domain, I perform average pooling of the output WordPiece embeddings for each word. Thus, the output of the semantic context encoder is this sequence of word-domain semantic embeddings. The $\text{BERT}_{\text{BASE}}$ model was pre-trained using long-form text data and is fine-tuned during *stage-2* of training as part of the prosody predictor. The prosody predictor performed better with a fine-tuned $\text{BERT}_{\text{BASE}}$ model compared to a frozen model.

## 6.4 Baselines

In order to evaluate the proposed method alongside other models in the literature, I introduce two sequence-to-sequence models to act as baselines. The first is Tacotron-2 (Shen et al., 2018), S2S for brevity. S2S is an attention-based model and is well known in the literature. However, the proposed method, CAMP, uses an explicit duration model. In order to separate the benefit of duration modelling and the core contribution of this chapter (i.e. using context to directly predict prosody), I use a second baseline: DURIAN+. DURIAN+ is very similar to S2S, but uses an explicit duration model, like it's progenitor, DurIAN (Yu et al., 2019). This allows the effects of duration modelling to be controlled for when evaluating CAMP.

### 6.4.1 S2S: Attention-based model

The current state-of-the-art TTS models consist of a phone encoder, attended over by an autoregressive acoustic decoder. Tacotron-2 (Shen et al., 2018) has been adopted as the prototypical S2S model in the literature, and is used as a baseline system in many recent studies.

S2S (i.e. Tacotron-2) is illustrated in Figure 6.4a. S2S uses the same phone encoder architecture as the *TTS model*'s phone encoder (Figure 6.2c, pp 141). S2S's acoustic decoder is autoregressive and uses 4 LSTM layers followed by a post-net of 5 CNN layers (Shen et al., 2018). The decoder attends over the phone embeddings using location-sensitive attention (Chorowski et al., 2015). The model has two losses using the reference mel-spectrogram, one with the LSTM output and one with the CNN post-net output.

### 6.4.2 DurIAN+: Explicit duration model

The mapping between an utterance's phones and its speech is monotonic. However, attention was developed for machine translation and its key design characteristic is the ability to align sequences non-monotonically (Bahdanau et al., 2014). There has been increasing interest in the literature in sequence-to-sequence models that only allow for monotonic alignments. For example, step-wise hard monotonic attention restricts the attention mechanism to be monotonic and attend to one phone at a time (Yasuda et al., 2019; He et al., 2019).

Attention serves two purposes in TTS: prediction of timing (i.e. alignment) and summarisation of relevant phonetic context. FastSpeech-2 (Ren et al., 2020) and DurIAN (Yu et al., 2019) demonstrated that: a duration prediction model can handle timing; and convolutions, self-attention, or bi-directional recurrent layers in the decoder can summarise local context. Since attention is not a requirement for state-of-the-art TTS, the approach taken here uses an explicit duration model, predicting durations used for upsampling.

Explicit duration S2S models (duration-based S2S) provide a bridge between attention-based models and SPSS models, but with the quality of current generation TTS. Duration-based S2S is equivalent to step-wise hard monotonic attention, except with explicit supervision of duration prediction. Some duration-based S2S models, such as DurIAN (Yu et al., 2019), use a duration predictor that is

trained separately from the phone encoder and acoustic decoder, this results in a model very similar to SPSS, but with different linguistic features, more layers, and a neural vocoder. Others, like FastSpeech-2 (Ren et al., 2020), jointly train the duration predictor's loss with the acoustic loss, making them more similar to attention: jointly learning to align and synthesise (Bahdanau et al., 2014).

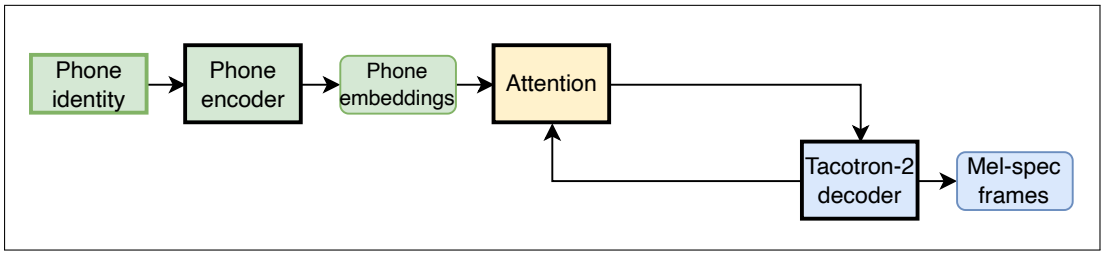
The second baseline, DurIAN+, is illustrated in Figure 6.4b. DurIAN+ brings together architectural choices from DurIAN, Tacotron-2, and FastSpeech-2. Like DurIAN, it uses a duration predictor instead of attention for alignment and upsampling. Like FastSpeech-2, the duration predictor is trained jointly with the phone encoder and acoustic decoder, and, unlike DurIAN, the duration predictor and acoustic decoder share a single phone encoder. Thus, the phone encoder is influenced by both the acoustic and duration losses. This means the phone embeddings used by the duration predictor are more informative in DurIAN+ than in DurIAN. DurIAN+'s phone encoder and acoustic decoder follow Tacotron-2's architecture. At the time of this work no other research had demonstrated a *significant improvement* making duration-based S2S state-of-the-art, hence my use of two baselines.

## 6.5 Experiments

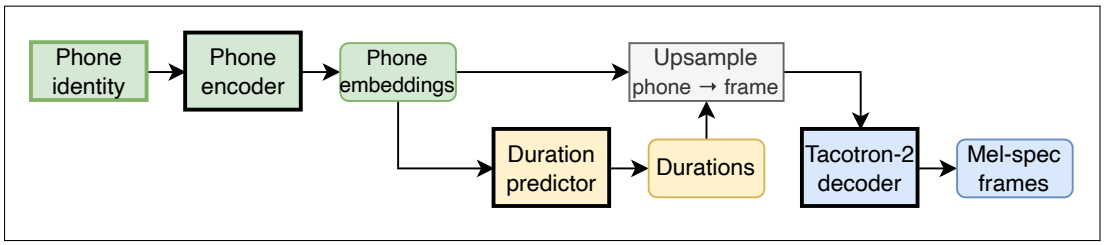
Here I introduce three evaluations, with the aim to demonstrate that CAMP is a state-of-the-art model, to understand the contribution of the proposed context features, and to determine the representational capacity of the controllable *TTS model*.

### 6.5.1 Data

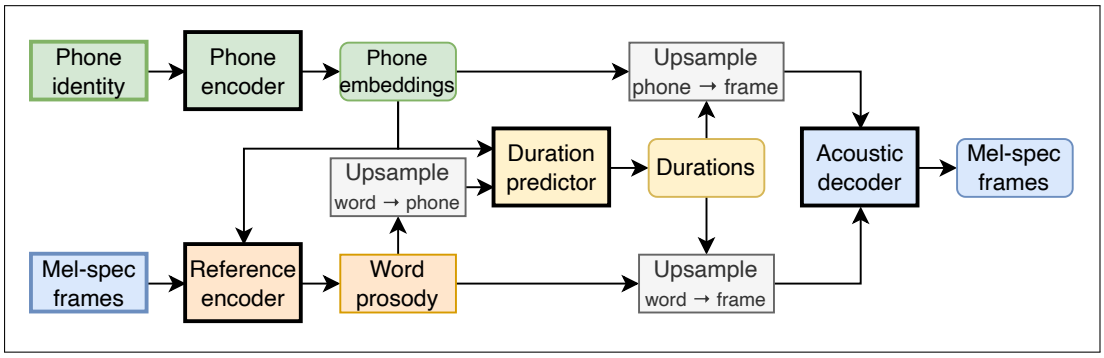
Models are trained on prosodically rich data: an expressive single-speaker proprietary Amazon dataset. The data contains about 25,000 utterances of professionally recorded long-form speech of a native US English female speaker. The utterances are an average of 16 words. The training, validation, and test sets are approximately 30 hours, 2 hours, and 6 hours, respectively. Phone features are one-hot encodings of phone identity, silences, word boundaries, and start or end of sentence tokens. Acoustic features are 80-band mel-spectrograms with a 12.5 ms frame-shift. Durations were extracted using forced alignment with Kaldi.



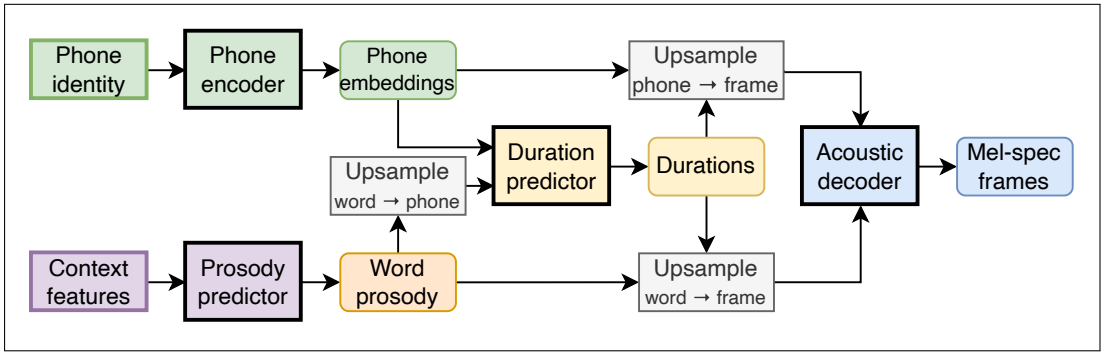

(a) S2S: Traditional S2S baseline using attention




(b) DurIAN+: S2S baseline using explicit durations




(c) ORA: Copies oracle prosody from human speech. This represents the top-line performance of the proposed system CAMP.




(d) CAMP: Proposed system using context features to predict prosody.

Figure 6.4: Architectures of evaluated systems.

### 6.5.2 Systems

Across the listening tests, I use 4 TTS systems, illustrated in Figure 6.4: S2S, DURIAN+, ORA, and CAMP. The differences between these systems are compared in Table 6.1. S2S is the attention baseline and DURIAN+ is the explicit duration baseline. ORA, uses oracle prosody extracted from reference human speech and represents the top-line performance that can be expected from my proposed two-stage approach. The proposed system, CAMP, uses context features to predict word-level prosody representations that to drive the *TTS model*. All models use the same autoregressive WaveNet vocoder that models each sample's conditional distribution with a mixture of logistics distribution (van den Oord et al., 2018). The vocoder synthesises waveforms with a sampling rate of 24 kHz and is trained on natural speech. NAT is natural 24 kHz speech without vocoding.

All models are trained using the Adam optimiser (Kingma and Ba, 2014). The acoustic and duration losses both use an $L1$ loss. ORA (i.e. *Stage-1*), S2S, and DURIAN+ are trained for 300,000 steps, with a learning rate of 0.001 and a decay factor of 0.98. *Stage-2* uses a Huber loss with $\rho = 1$ to train the prosody predictors. The Huber loss is a combination of the $L1$ and $L2$ losses. It is quadratic below $\rho$ and linear above $\rho$, making it less sensitive to outliers than

Table 6.1: Comparison of differences between evaluated systems. S2S is an attention-based baseline. DURIAN+ is an improved baseline using explicit duration modelling. ORA represents my approaches' best-case performance. CAMP is my proposed model using context information to predict prosody.

| | S2S | DURIAN+ | ORA | CAMP |
|---|---|---|---|---|
| Phone encoder | Tacotron-2 encoder (Fig. 6.2c) | | | |
| Prosody representation | — | — | Reference encoder (Fig. 6.2a) | Prosody predictor (Fig. 6.3a) |
| Duration predictor inputs | — | Phone embeddings | Phone embeddings and prosody representation | |
| Acoustic Decoder | Tacotron-2 decoder | | Non-autoregressive decoder (Fig. 6.2b) | |

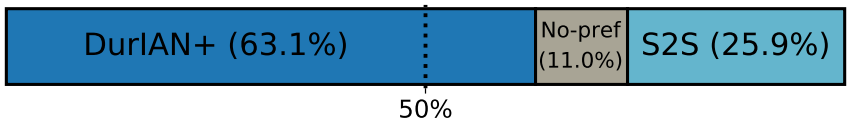
squared error losses. This produced marginally better prosody than an $L1$ or $L2$ loss during informal listening. Prosody predictors are trained for 100,000 steps, with a learning rate of 0.0001 and a decay factor of 0.98.

The hyperparameters in *stage-1* of training needed tuning to ensure the representation learnt was suitably disentangled from phonetic content, speaker identity, and background noise. The number of dimensions for the word-level representation was tuned, and different conditioning features were tried for the reference encoder, such as phone identity features and phone embeddings. To judge linguistic disentanglement of the representations, I synthesised speech using a sequence of words assembled from two utterances with the oracle prosody from the first utterance. This demonstrated how robust the representations are to different linguistic content. To judge what prosodic information the representations captured, I synthesised speech using a sequence of word-level prosody representations assembled from two utterances with the words from the first utterance. This demonstrated how much the representations can control the prosody for fixed linguistic content. Through informal listening with the output from these two modes of synthesis, I was able to tune *stage-1* before developing and training *stage-2*.

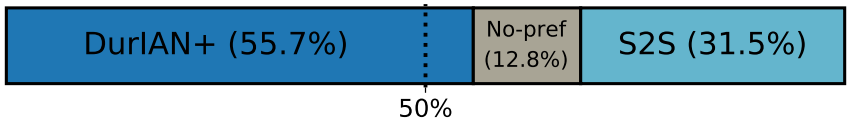
### 6.5.3 Subjective evaluation

To evaluate CAMP, first we need to determine which is the stronger baseline, S2S or DURIAN+, and what context features are most impactful for CAMP. Thus, three listening tests were performed. The first investigates the contribution of duration modelling, and is used to select the stronger baseline. The second is an ablation study to determine the contribution of the proposed syntactic and semantic context features. Finally, the main evaluation measures listener preference of the proposed CAMP system by comparing it with: a baseline, ORA, and NAT.

The listening tests all use the same 96 sentences. These were chosen randomly from the 6 hour test set. Listeners are all native US English speakers, were vetted (as opposed to being crowd-sourced), were paid for participation, and used headphones. Other demographic information of listeners is not available.




(a) Full pairwise preference results on all 96 test sentences.




(b) Pairwise preference results for the 76 sentences that exhibited no attention-related instabilities. This demonstrates that duration modelling improves the quality, not just robustness.

Figure 6.5: Preference test demonstrating that a jointly-trained duration predictor improves significantly over Tacotron-2. Question asked: *"Choose which version you prefer"*. (a) and (b) show results from the same listening test, with (b) reporting only sentences without attention-related instabilities.

#### 6.5.3.1 Benchmark model

During informal listening, it was found that DurIAN+ produced more natural speech, with fewer artefacts, and improved prosody. Both for this reason, and to control for the contribution of explicit duration modelling in CAMP, DurIAN+ should be used as the baseline when evaluating CAMP. To formally verify the apparent improvement over S2S, I present a listening test that benchmarks DurIAN+ against S2S.

I perform a preference test to directly compare S2S and DurIAN+. The test includes a "no preference" option as the similarity of the two system's design may result in stimuli that are difficult to tell apart. The preference test was completed by 15 listeners. Participants were asked to "Choose which version you prefer".

The results, shown in Figure 6.5a, were tested using a binomial significance test. This test confirmed that DurIAN+ is significantly preferred over S2S, with very high confidence: $p < 10^{-15}$. In particular, DurIAN+ is overwhelmingly preferred for some sentences. These were sentences where S2S produced artefacts related to known issues with attention. There were no sentences where DurIAN+ had stability issues. By design, S2S models using an explicit duration model will not suffer from attention-related instability and are overall more robust to

synthesis failure modes and articulation mistakes.

To evaluate if DurIAN+ improves on aspects other than robustness, the results are filtered to only include sentences that do not suffer from signal artefacts. There are 20 sentences with such issues for S2S. The filtered results are presented in Figure 6.5b. A binomial significance test finds DurIAN+ to be significantly preferred with $p < 10^{-15}$. This demonstrates that DurIAN+ greatly improves listener preference, i.e. naturalness or prosody quality, not just the system's robustness.

While other studies have shown that duration modelling does not degrade performance (Yu et al., 2019; Ren et al., 2020), this result was the first result in the literature to demonstrate a significant improvement attributable to duration modelling. An ablation was not performed to determine exactly what led to this improvement compared to other approaches using explicit durations. However, I hypothesise it is due to the use of a shared phone encoder and joint training of duration and acoustic losses. These are design choices not consistently used in prior works.

Tacotron-2 has been widely compared with other approaches in the literature. These results transitively link the final evaluation of CAMP (Section 6.5.3.3) with such results in the literature.

#### 6.5.3.2 Context feature ablation

In Section 6.3.2, I introduced 5 context features: fine-tuned word-level BERT embeddings; and, 4 syntactic context features—part of speech, word class, compound noun structure, and punctuation structure. Here, I evaluate the contribution of these context features to the prosody predictor. Three versions of CAMP were trained, each uses a different set of context features for the prosody predictor:

$\text{CAMP}_{\text{Syntax}}$ — Prosody predictor trained with 4 syntax context encoders.

$\text{CAMP}_{\text{BERT}}$ — Prosody predictor trained with the BERT context encoder.

$\text{CAMP}_{\text{BERT+Syntax}}$ — Prosody predictor trained with all 5 context encoders.

These three models each use a separately trained prosody predictor, but all models use the same frozen *TTS model* and are trained to predict the same word-level prosody representations.

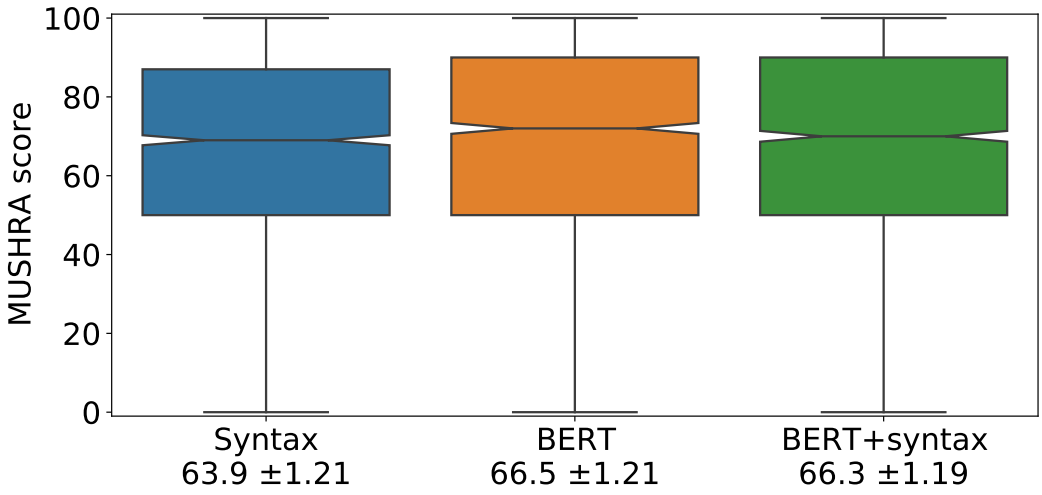



Figure 6.6: MUSHRA listening test results for ablation of context features in CAMP. Question asked: *"Rate the systems based on your preference"*. Mean rating and 95% confidence intervals are reported below system names. The middle line of each box represents the median rating, the notch represents the confidence interval around the median.

To determine the contribution of each to user preference, I conduct an ablation test using a MUSHRA-like format (BS Series, 2014). The stimuli are isolated utterances, meaning there is no concrete concept of a correct prosodic rendition (Clark et al., 2019). Thus, in this evaluation, no systems are used as a reference or anchor in the MUSHRA design. Listeners were not required to rate any systems as 0 or 100, but were free to do so. The listening test was completed by 20 listeners. Listeners were asked to "Rate the systems based on your preference" on a scale from 0 to 100.

The results in Figure 6.6 show that $\text{CAMP}_{\text{Syntax}}$ has a lower mean rating than the two systems using BERT. I perform two-sided Wilcoxon signed-rank tests on all 3 pairs of systems, and correct with Holm-Bonferroni. These tests find that $\text{CAMP}_{\text{Syntax}}$ is indeed significantly worse than the other two models ($p < 0.001$). No statistically significant difference is found between $\text{CAMP}_{\text{BERT}}$ and $\text{CAMP}_{\text{BERT+Syntax}}$ ($p = 0.83$).[2] Confidence intervals, reported in Figure 6.6, are computed using percentile bootstrap (Davison and Hinkley, 1997).[3]

[2]The ratings were also converting into rankings and analysed using non-parametric tests, this provided the same statistical conclusions.

[3]Percentile bootstrap is a non-parametric technique that uses sampling with replacement to

It is clear that using BERT leads to a significant improvement in preference. The results also suggest that the syntactic features provide no additional information compared to BERT's contextualised representations. This is in agreement with findings that BERT can capture both semantic *and* syntactic information (Rogers et al., 2020). As such, I use CAMP$_{\text{BERT}}$ as the proposed TTS system.

#### 6.5.3.3 Prosody modelling evaluation

Finally, we turn to the proposed system. Here I compare CAMP$_{\text{BERT}}$ with three systems: DURIAN+, ORA, and NAT. DURIAN+ serves as a baseline, as it does not include any improvements to prosody modelling. ORA is the top-line for my proposed approach, i.e. using the best-case prediction of the prosody representation. And, NAT is the reference: human speech. Choosing not to use vocoded speech as the reference means this evaluation will also assess the quality of the vocoder. This removes the need for an additional listening test demonstrating the quality of the vocoder. This decision was made as the vocoder was very high quality.

By using DURIAN+ as the baseline, any improvement observed in CAMP$_{\text{BERT}}$ over this baseline can be attributed to the two-stage design. While it may seem inconsistent to compare two systems that use different inputs—DURIAN+ does not use additional context features—I did experiment with adding additional context to DURIAN+ (and S2S) and observed no clear improvement. Specifically, I added a pre-trained BERT$_{\text{BASE}}$ model with word-level average pooling as an additional encoder, parallel to the phone encoder. The word-level semantic embeddings were upsampled to phone-level and concatenated with the phone embeddings. However, in informal listening tests, this yielded no noticeable improvement in prosody, similar to previous findings (Hayashi et al., 2019; Fang et al., 2019). This is likely due to DURIAN+ predicting the frame-level mel-spectrogram, as opposed to directly predicting prosody representations. This means that the BERT$_{\text{BASE}}$ model is fine-tuned for an acoustic modelling task, making it less likely to improve the prosody. Due to this, the version of DURIAN+ without additional input was used for simplicity, and to allow easier comparison with S2S and other work in the literature as discussed.

The four systems, shown together in Figure 6.4 (pp. 150), are evaluated using

simulate many virtual experiments from which confidence intervals can be approximated. For a more detailed overview and tutorial see Rousselet et al. (2021).

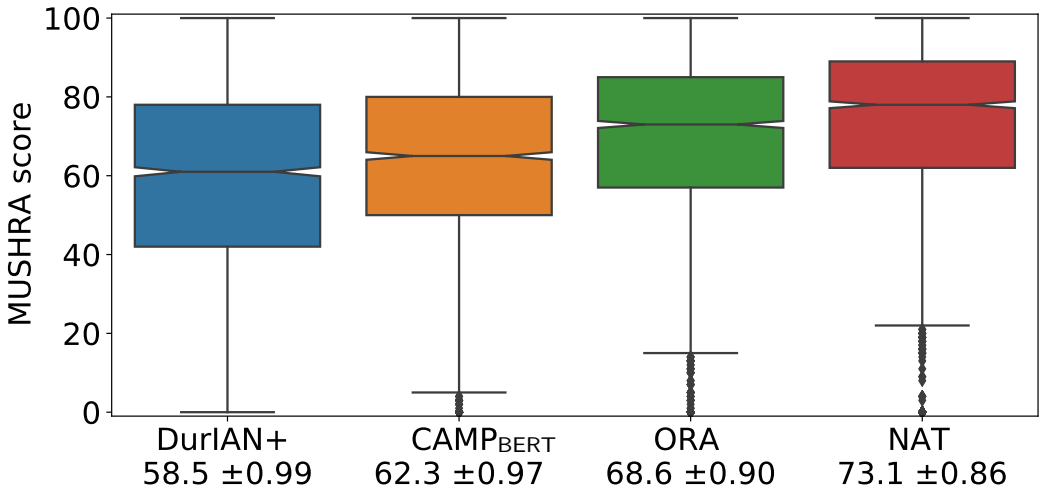



Figure 6.7: MUSHRA listening test results with CAMPBERT. Question asked: *"Rate the systems based on your preference"*. Mean rating and 95% confidence intervals are reported below system names. Diamonds represent outliers.

a MUSHRA-like test (BS Series, 2014). Like in the previous test, the stimuli are isolated utterances, so there is no concept of a correct prosodic rendition (Clark et al., 2019). Thus, no systems are used as a hidden reference or anchor. Listeners were not required to rate any systems as 0 or 100, but were free to do so. 25 listeners completed this test. The listeners were asked to "Rate the systems based on your preference" on a scale from 0 to 100.

A box plot of the MUSHRA results is presented in Figure 6.7, this includes 95% confidence intervals, computed using percentile bootstrap. I perform two-sided Wilcoxon signed-rank tests for all 6 pairs of systems, and correct with Holm-Bonferroni. All systems are significantly different from each other with a very high level of confidence: $p \ll 10^{-10}$.[4] The results show that CAMPBERT is significantly better than the baseline, DurIAN+, but significantly worse than the top-line, ORA, and natural speech. Overall, NAT was significantly preferred compared to all other systems. However, NAT was rated relatively low in some cases, as illustrated by the outliers in Figure 6.7. This was due to the high quality of the synthetic speech produced by the other systems. Additional analysis of this is presented in Appendix C.

[4]I also converted the ratings to rankings and performed non-parametric statistical tests, this provided the same statistical conclusions. See Figure C.1 (pp. 178) for the rank-based results.

Table 6.2: Three gap reduction comparisons made using the MUSHRA results. Values in parentheses are the mean ratings from the MUSHRA results reported in Figure 6.7. Gap reduction is defined as the relative position (as a percentage) of the system of interest compared to the lower-bound and upper-bound systems. Confidence intervals are computed using percentile bootstrap.

| | System | | Lower-bound | Upper-bound | Gap reduction |
|---|---|---|---|---|---|
| #1 | CAMP$_{\text{BERT}}$ | (62.3) | DURIAN+ (58.5) | NAT (73.1) | $26\% \pm 7\%$ |
| #2 | ORA | (68.6) | DURIAN+ (58.5) | NAT (73.1) | $69\% \pm 6\%$ |
| #3 | CAMP$_{\text{BERT}}$ | (62.3) | DURIAN+ (58.5) | ORA (68.6) | $38\% \pm 10\%$ |

The high degree of confidence confirms that the large differences of mean ratings in Figure 6.7 are meaningful, i.e. it isn't due to high variance. However, this doesn't illustrate size of the improvement found. Compared to a lower-bound system, we can look at how close a system comes to reaching an upper-bound system. I refer to this as the gap reduction, i.e. how much does a system of interest reduce the gap from a lower-bound to an upper-bound. This is a relative measure of the distance between the lower-bound and upper-bound. Three such gap reduction comparisons are presented in Table 6.2, along with confidence intervals on the gaps, computed with percentile bootstrap. Comparison #1 reveals how close the proposed approach comes to reaching natural speech, while comparisons #2 and #3 illustrate the performance of *stage 1* and *stage 2*, respectively.

The most important result is comparison #1: CAMP$_{\text{BERT}}$ is able to get 26% of the way to parity with natural speech. This is with respect to a strong baseline, DURIAN+, which was shown to be an improvement over Tacotron-2. Improving the learnt prosody representation or adding more context information could move CAMP$_{\text{BERT}}$ closer to NAT. The two other comparisons probe the efficacy of the reference encoder and the prosody predictor.

Comparison #2 provides insight on the best-case performance of CAMP$_{\text{BERT}}$, i.e. if the prosody predictor performed perfectly. This shows that the learnt representation is capable of closing the gap to natural speech by 69%. This means the representation is lossy, likely due to the trade-off between disentanglement and descriptive power, controlled through the temporal and dimensional bottlenecks. While ORA can be improved through architecture changes, it is likely to never

reach 100% as complete disentanglement of prosody and phonetics may not be possible.

Finally, comparison #3 sheds light on the use of context features for predicting prosody. This shows that a fine-tuned BERT model provides enough context to capture 38% of the prosodic behaviour, for this model and data. Intuitively, we should expect to see diminishing returns when pushing this gap reduction closer to 100%.

### 6.5.4 Discussion

As discussed above, the use of additional context information alone may not improve prosody modelling. The single-stage models, S2S and DURIAN+, did not improve noticeably when adding BERT as an additional encoder. I attribute the performance of $\text{CAMP}_{\text{BERT}}$ to its use of a prosodically-relevant loss, combined with the introduction of more context information. The prosody predictor's loss must focus directly on prosody, not frame-level spectrogram targets. In this work, I achieve this by predicting a disentangled prosodic representation. However, the same effect could be achieved by predicting explicit prosody features, such as $F_0$, as explored by Ren et al. (2020). Any inductive bias that encourages the model to utilise context information for slower-varying aspects of speech (e.g. prosody), should be able to achieve a similar effect as seen here. This could mean using a self-supervised loss, such as contrastive learning (Baevski et al., 2020), designed specifically to focus on prosody.

While my results show a large improvement in prosody quality, there is clearly room for more progress within this two stage paradigm. Improved representation learning models and disentanglement could raise the best-case performance seen with ORA in comparison #2 (Table 6.2). As for the prosody predictor, there are two avenues to improve prosody prediction: using more context features, especially by considering what prosodic variation occurs in the data (explored in Chapter 5); and using wider context either by providing the prosody predictor with surrounding utterances or by training the prosody predictor on longer extracts of text, such as turns or paragraphs.

I experimented briefly with training the prosody predictor on paragraphs. While this improved the prosody at utterance boundaries, the paragraph-level

model only made significant prosodic changes for short utterances within a paragraph. This is likely due to the use of LSTMs in the prosody predictor, which have a limited receptive field. This limitation was of less concern when modelling sentences, but for longer sequences replacing recurrent layers in the prosody predictor with self-attention may allow the model to better exploit more distant context information. A model with the capability of bidirectional generation would be another interesting direction to explore (Lawrence et al., 2019).

These experiments used single-speaker data. This makes the task much more manageable, as different speakers exhibit different prosodic behaviours. However, moving to a multi-speaker prosody representation might lead to better disentanglement. More interestingly, investigating a multi-speaker prosody predictor might provide insights into the relationship between different speakers' prosodic patterns.

## 6.6 Conclusion

I introduced CAMP, a two-stage approach for prosody modelling. In *stage-1*, a prosody representation is learnt from the mel-spectrogram using a novel word-domain reference encoder. In *stage-2*, a prosody predictor is trained that uses context features to predict the disentangled word-domain representations learnt in *stage-1*. This approach is able to close the gap between a strong state-of-the-art baseline and natural speech by 26%.

There are two main contributions of my approach: incorporating additional context, and directly modelling prosody. By adding new context information the model will be more able to make meaningful prosodic decisions (**Theme 3**), instead of producing average prosody. I also stress the importance of the latter contribution; adding inductive bias that focuses the system on prosody modelling is imperative. Without this inductive bias, the additional context features will be used to predict frame-level detail in the spectrogram, as opposed to suprasegmental prosody. Such an inductive bias was achieved in CAMP using a prosodically-relevant loss.

Additionally, I presented an intermediate result demonstrating that replacing attention with a duration model led to significantly better performance. Based on design choices included and omitted in different prior work, I hypothesise

that joint training of the duration model and the use of a shared phone encoder contribute to this improvement. Work conducted at the same time by Shen et al. (2020) corroborates these results. Shen et al. (2020) go further and propose novel changes to explicit duration modelling, most notably Gaussian upsampling, which can be seen as a parallel to attention's weighted summarisation.

# Chapter 7

# Conclusion

This thesis developed approaches to synthesise multiple prosodic renditions from a fixed input: random sampling, human control, and learnt representations. As outlined by this thesis's claim, either the context should determine which prosodic rendition is appropriate, or when there is insufficient context, prosody must be controlled or randomly sampled.

> *Appropriate prosody can be synthesised with insufficient context, but prosodic variation not determined by the available context must be controlled by a human or modelled probabilistically.*

Chapter 3 demonstrated that in TTS, prosody is one-to-many and that if this is not taken into account, the synthesised prosody will be monotonous. Chapters 4 and 5 provided methods that make human-in-the-loop control faster and more intuitive. Finally, in Chapter 6, I proposed a state-of-the-art model that incorporates additional context to predict appropriate prosody.

While state-of-the-art S2S models and neural vocoders can produce synthetic speech with excellent acoustic quality, it is clear from experiments in this thesis, and in the literature, that prosody in synthetic speech is still lacking in appropriateness. Through the three themes outlined in Chapter 1, I explored several core challenges facing prosody synthesis: prosody is embedded in speech alongside segmental content and speaker identity, we have no clear orthography for prosody, and we lack sufficient context information to predict prosody.

**Theme 1**, controllability, provides a solution to prosody's entanglement in speech, either through explicit features, like $F_0$ and duration, used in Chapters 3

and 5, through labelled control explored in Chapter 4, or through learnt, disentangled representations developed in Chapter 6.

Prosody models that can produce multiple renditions can be controlled by human operators, however it's important that the control interface is usable. Since there is no clear orthography for prosody, other options must be explored to make control usable. I investigated this through interpretability (**Theme 2**). In Chapter 4, I exploited found data to augment the TTS dataset with interpretable labels. While in Chapter 5, I used unsupervised learning with inductive biases constructed to learn discrete categories. These were found to produce distinct prosodic behaviours that were interpretable, although not consistent.

Finally, in **Theme 3**, I considered the use of context for predicting appropriate prosody. In Chapter 6, I utilised pre-trained foundation models to extract useful semantic and syntactic information. My proposed approach can extend to include more types of context as it becomes available, both new context features and surrounding context. Importantly, I observed that additional context should be used for *prosody* prediction, to ensure that the causal information is used efficiently—compared to using it for *spectrogram* prediction, as seen in other recent research.

## 7.1 Future work

Evaluation of prosody is challenging. While unrealistic prosody can be identified with current methods, measuring appropriateness is more difficult due to the need for context. Clark et al. (2019) suggested that using neighbouring utterances leads to higher ratings, but O'Mahony et al. (2021) showed the wording of the test was a confound: when evaluating naturalness, the context does not impact ratings. However, listeners do interpret appropriateness as a different concept than naturalness (O'Mahony et al., 2021). Wallbridge et al. (2021) demonstrated that context is important for evaluating appropriateness.

Fortunately, the field is beginning to focus on developing evaluations of prosody for TTS. For example, prosody is normally evaluated at the utterance level, but Gutierrez et al. (2021) demonstrated that evaluating prosody in the word domain can provide more information than sentence-level MOS ratings, finding that most prosodic errors preceded punctuation. Since prosody is so in-

tertwined with context, perhaps we should categorise our models' error modes using a fine-grained evaluation such as this, and determine what design choices or context features would address each failure mode.

I explored different methods for prosody control independently in this thesis: prosodic sampling, human control, and context-based prediction. A promising direction for future work would be to combine these. A conditional prosodic distribution—defining the set of appropriate choices—could be predicted using context. This conditional distribution could be further conditioned using human-driven control inputs. Finally, a single prosodic rendition could be sampled. This approach brings together the different solutions required to handle prosody depending on the available context, as illustrated in Figure 1.1 (pp. 4).

The limited use of context in existing models is related to the constraints of current machine learning techniques. With advances in representation learning (Le-Khac et al., 2020; Schölkopf et al., 2021) and graph neural networks (Battaglia et al., 2018; Wu et al., 2020b), more types of context can be experimented with and incorporated. While data collection of context information remains a bottleneck, training with wider context using state-of-the-art techniques, like self-attention (Vaswani et al., 2017) or memory networks (Santoro et al., 2016; Borgeaud et al., 2021), might allow context to be accessed directly from speech data, instead of needing explicit context features. Recent work has demonstrated the use of such advances: Karlapati et al. (2021) used graph neural networks to utilise constituency parse trees, and Oplustil-Gallegos et al. (2021) used representation learning to incorporate context from surrounding text and acoustics.

With recent developments in machine learning, it may be possible to generate appropriate prosody implicitly—without explicit context features—by using surrounding context. In computer vision, content and style can be disentangled using inductive bias in the model design with no content or style labels (Huang et al., 2018), but control is limited to the styles observed in the data. To control prosody, the equivalent data requirement is to use longer extracts of speech, i.e. the surrounding context. Powerful natural language generative models can produce output with coherent topics (Brown et al., 2020), but training such models requires even longer extracts of text, e.g. entire documents. To generate coherent prosody without explicit context features, a single training data point may

need to be vastly longer than a single utterance, such as a full conversation, book chapter, or news article. In natural language processing, high-level structure such as grammar can be learnt implicitly (Rogers et al., 2020) by leveraging very large data. To make use of the surrounding context and learn interesting prosodic structure, we may require a similarly large quantity of data. Fortunately, there exist datasets with over 50,000 hours of speech (Abu-El-Haija et al., 2016; Clifton et al., 2020; Kahn et al., 2020). Working with longer extracts and larger data would require new training methods, such as new self-supervised losses (Yamaguchi et al., 2021) and better inductive biases (Baevski et al., 2020) from the emerging topic of foundation models (Bommasani et al., 2021). Generating appropriate prosody based on surrounding context is a rich research direction, but a very challenging one.

I investigated prosody for single speakers in a single language. Understanding how prosody varies across speakers will make new techniques useful in more TTS applications. Additionally, synthesising appropriate prosody for languages other than English would make speech technology available to more users.

*This work was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.*

# Appendix A

# Features

## A.1 SPSS linguistic features

Linguistic features used for SPSS models in Chapters 3 and 4 are detailed in Table A.1.

## A.2 eGeMAPS emotion features

The eGeMAPS features used to predict emotion labels in Chapter 4 are detailed in Table A.2.

Table A.1: Linguistic features used in Chapters 3 and 4.

| |
|---|
| Quin-phone |
| Phone identity before the previous phone |
| Previous/current/next phone identity |
| phone identity after the next phone |
| ToBI prediction |
| ToBI endtone of the current phrase |
| Syllable structure |
| Name of the vowel of the current syllable |
| Whether the previous/current/next syllable stressed or not |
| Whether the previous/current/next syllable accented or not |
| Number of stressed syllables before/after the current syllable in the current phrase |
| Number of accented syllables before/after the current syllable in the current phrase |
| Position of the current syllable between the previous and next stressed syllables (forward/backward) |
| Position of the current syllable between the previous and next accented syllables (forward/backward) |
| POS structure |
| Guessed part of speech (gpos) of the previous/current/next word |
| Number of content words before/after the current word in the current phrase |
| Position of the current word between the previous and next content words (forward/backward) |
| Utterance structure |
| Position of the current phone in the current syllable (forward/backward) |
| Position of the current syllable in the current word/phrase (forward/backward) |
| Position of the current word in the current phrase (forward/backward) |
| Position of the current phrase in utterence (forward/backward) |
| Number of phones in the previous/current/next syllable |
| Number of syllables in the previous/current/next word/phrase |
| Number of words in the previous/current/next phrase |
| Number of syllables/words/phrases in this utterence |

Table A.2: eGeMAPS low-level descriptor features.

| **1 energy related LLD** | **Group** |
|---|---|
| Loudness (signal intensity) | Prosodic |
| **25 spectral LLD** | **Group** |
| $\alpha$ ratio - 50-1000 Hz & 1000-1500 Hz | Spectral |
| Spectral slope - 0-500 Hz & 500-1500 Hz | Spectral |
| Hammarberg index | Spectral |
| MFCC 1-4 | Cepstral |
| Spectral flux | Spectral |
| **16 voicing related LLD** | **Group** |
| Log $F_0$ on a semi-tone scale | Prosodic |
| Formant 1-3 frequency | Voice quality |
| Formant 1-3 bandwidth | Voice quality |
| Formant 1-3 amplitude | Voice quality |
| Harmonic difference - H1-H2 & H1-A3 | Voice quality |
| Harmonics-to-noise ratio | Voice quality |
| Jitter of consecutive $F_0$ periods | Voice quality |
| Shimmer of consecutive $F_0$ periods | Voice quality |

# Appendix B

# Full results for Chapter 5

## B.1 Pairwise preference results

The full pairwise result are reported for $\text{AE}_{\text{K–MEANS}}$ in Figure B.1, and for $\text{VAE}_{\text{VAMP}}$ in Figure B.2.

## B.2 Descriptive terms

The list of 12 sentences used for the qualitative evaluation can be found in Table B.1. Descriptive terms used more than once for these sentences are given in Table B.2, and descriptive terms used only once are given in Table B.3.

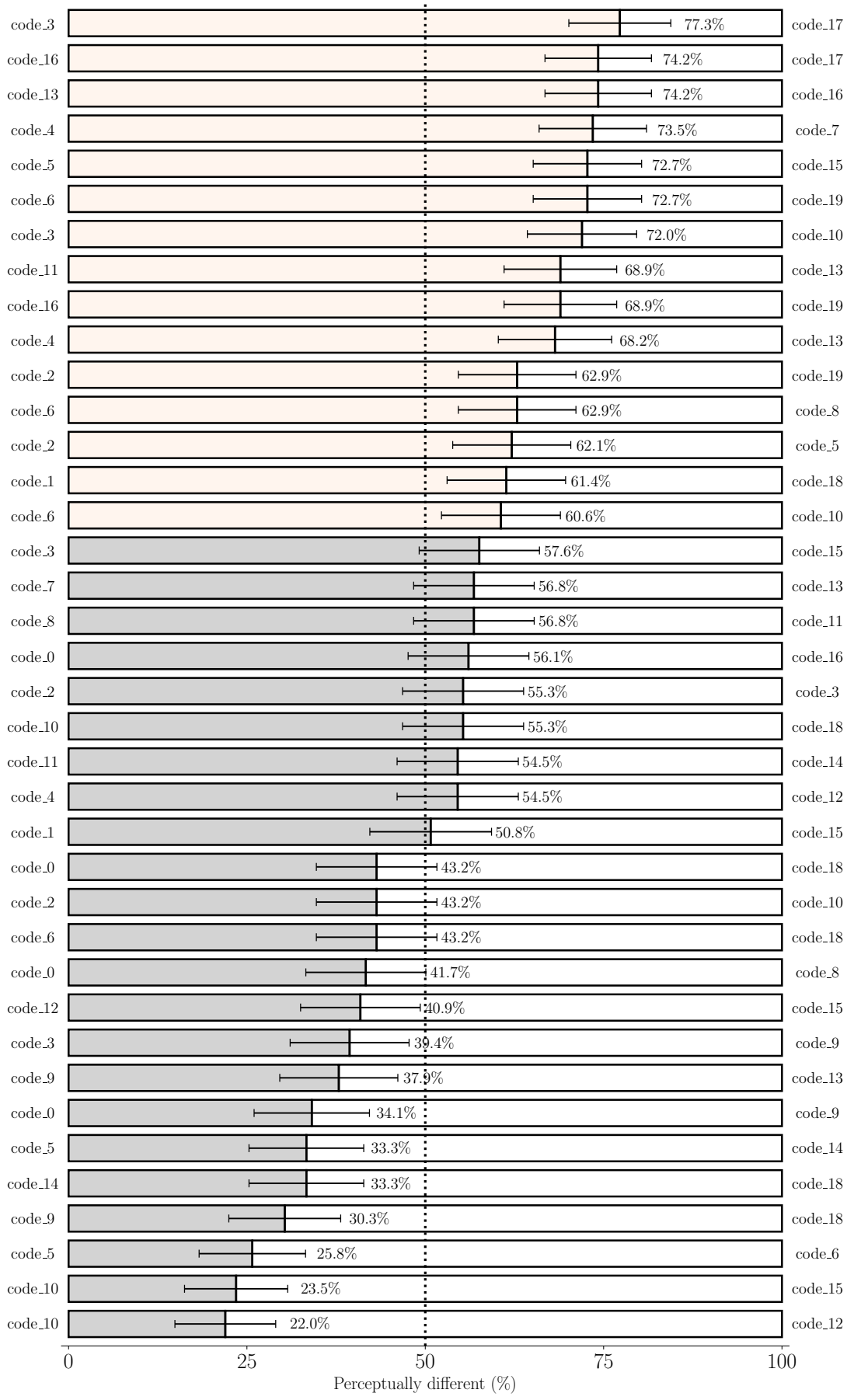



Figure B.1: Same/different results for all 36 intonation code pairs in $\mathrm{AE}_{\mathrm{K\text{–}MEANS}}$. Error bars shows binomial confidence intervals.

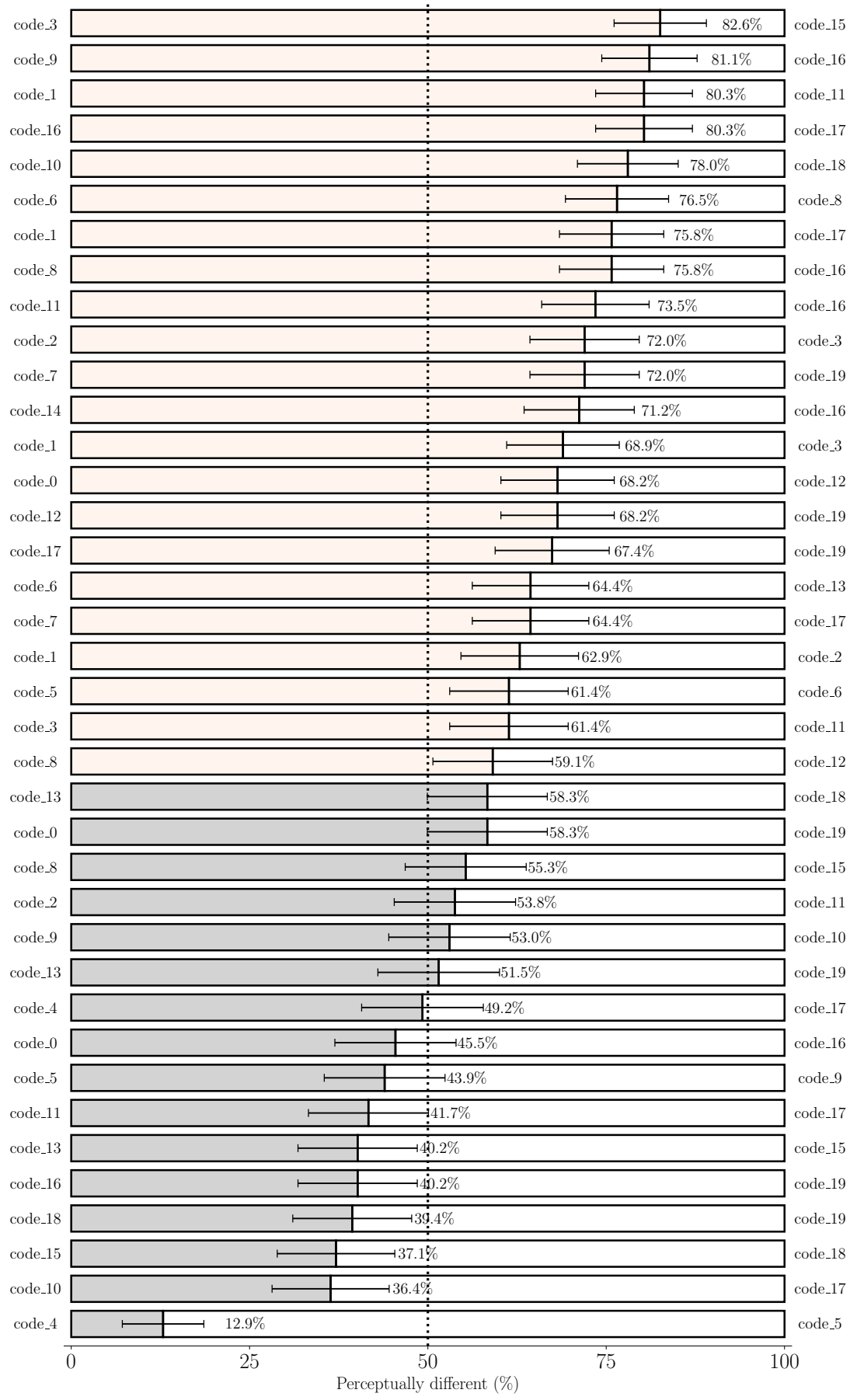



Figure B.2: Same/different results for all 36 intonation code pairs in $\mathrm{VAE}_{\mathrm{VAMP}}$. Error bars shows binomial confidence intervals.

Table B.1: Abbreviations used in Tables B.2 and B.3 for the 12 test sentences from the qualitative evaluation in Chapter 5.

| Abbreviation | Sentence |
|---|---|
| S1 | **There was no answer.** |
| S2 | **“I’m so hungry.”** |
| S3 | **“Too hard!”** |
| S4 | **They climbed the stairs.** |
| S5 | **“What’s the matter now?”** |
| S6 | **“We’d better make sure.”** |
| S7 | **“Do you think we’re so stupid?”** |
| S8 | **“I’m sorry.”** |
| S9 | **He wanted a turnip.** |
| S10 | **They both tugged and tugged.** |
| S11 | **But the turnip didn’t move.** |
| S12 | **“It’s enormous!” cried Jack.** |

Table B.2: Counts of descriptive terms that were used more than once. For abbreviations and full sentences, see Table B.1.

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | S11 | S12 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Upset | 3 | 4 | 1 | 2 | | 2 | 4 | 2 | | 3 | 4 | | 25 |
| Statement | 4 | 2 | 2 | 1 | 2 | 1 | | | 6 | | 4 | | 22 |
| Narrative | | 1 | | 2 | | | | | 6 | 6 | 2 | 1 | 18 |
| Question | 1 | 1 | 3 | | 2 | 2 | 1 | 5 | 1 | | | | 16 |
| Surprised | 2 | | 1 | | 1 | | | 1 | 1 | | 2 | 6 | 14 |
| Standard | 1 | | 1 | | | 2 | 1 | 3 | 1 | 2 | | 1 | 12 |
| Continuation rise | 1 | 2 | | 2 | | 1 | | 1 | 2 | 1 | | | 10 |
| Emotional | 1 | | 1 | | 1 | | 1 | 1 | | 1 | 1 | 1 | 8 |
| Fake apology | | | | | | | | 8 | | | | | 8 |
| Anticipatory | 2 | 1 | | 2 | 1 | | | | | | 1 | | 7 |
| Sad | 1 | | | 2 | | | 2 | | 2 | | | | 7 |
| Child storytelling | | | | | | 1 | | | | 3 | 1 | 1 | 6 |
| Insulted | | | | | | | 5 | 1 | | | | | 6 |
| Monotonous | 1 | | 1 | | 1 | 1 | | | 1 | | | | 5 |
| Rhetorical | | | | | 2 | | 3 | | | | | | 5 |
| Exclamation | | | 1 | | | | | | | | | 4 | 5 |
| Passive aggressive | | | | | | | | 5 | | | | | 5 |
| Confused | 1 | | | 1 | | | | 1 | | | 1 | | 4 |
| Bored | 1 | | | | | | | | 2 | | 1 | | 4 |
| Uncertain | | | | 1 | | 2 | | 1 | | | | | 4 |
| Apology | | | | | | | | 4 | | | | | 4 |
| Empathy | | | | 1 | | 1 | 1 | | | | | | 3 |
| Childlike | | | 1 | | | | | | | | | 2 | 3 |
| Annoyed | | | | 2 | | | 1 | | | | | | 3 |
| Friendly | | 1 | | | 2 | | | | | | | | 3 |
| Resigned | | 1 | 1 | | | | | | | | | | 2 |
| Grumpy | | | 1 | 1 | | | | | | | | | 2 |
| Disappointed | | | | | 1 | | | | | | 1 | | 2 |
| Urgent | | | | | 2 | | | | | | | | 2 |
| #Terms | 12 | 8 | 11 | 9 | 12 | 8 | 8 | 14 | 9 | 6 | 10 | 7 | |

Table B.3: Additional descriptive terms that were only used once. For abbreviations and full sentences, see Table B.1.

| Abbreviation | Descriptive terms |
|---|---|
| S1 | Distracted |
| S2 | Expressive, angry, animated |
| S3 | Dejected, stubborn, uninterested, determined |
| S4 | Assertive |
| S5 | Back-channelling, motherly, direct speech, concern, fed up, agitated, frustrated, moody |
| S6 | Cautious, hesitant, certain |
| S7 | Patronised, mature, self-assured, threatened |
| S8 | Humorous sarcasm, rant, sarcastic, unfriendly, disbelief |
| S9 | |
| S10 | Overdone |
| S11 | Overwrought |
| S12 | Scared, fake impressed, worry, unsurprised, excited, pleased, happy |

# Appendix C

# Additional analysis for Chapter 6

## C.1 Analysis of MUSHRA results

When evaluating the proposed CAMP$_{\text{BERT}}$ system, the MUSHRA results in Figure 6.7 (pp. 157) had a number of outliers for some systems. To further understand these results I conducted two further analyses of the same results.

Some listeners may use lower or higher average ratings for the same system, or may use relative position in different ways. To control for these differences, I converted the ratings, on a scale of 0 to 100, to rankings. The rank-based results, from rank 1 to 4, are plotted in Figure C.1. This illustrates that all systems were rated as most and least preferred in some stimuli. This means the synthetic speech systems are preferred to NAT in some cases. A non-parametric statistical test found the same statistical inferences as in Chapter 6: all system pairs are significantly different from each other.

Some listeners did not use the full range when rating systems, while others did. Listeners were not required to use the full scale. To control for this difference in behaviour, in Figure C.2 I plot normalised ratings. Normalisation was performed per-screen: the 4 ratings provided for a single screen are scaled linearly such that the lowest rating is 0 and the highest rating is 100. This form of normalisation crudely mimics a test where listeners were required to use the full range of the scale. For the same statistical tests as in Chapter 6, the same statistical inferences were found with the normalised ratings. This demonstrates that this difference in listener behaviour did not impact the results.

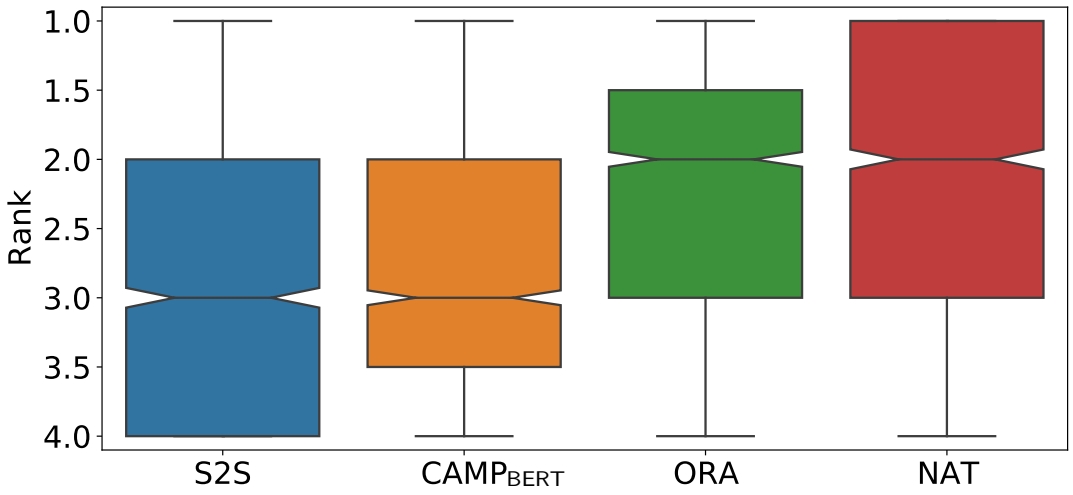



Figure C.1: MUSHRA listening test results by ranking with CAMP$_{\text{BERT}}$.

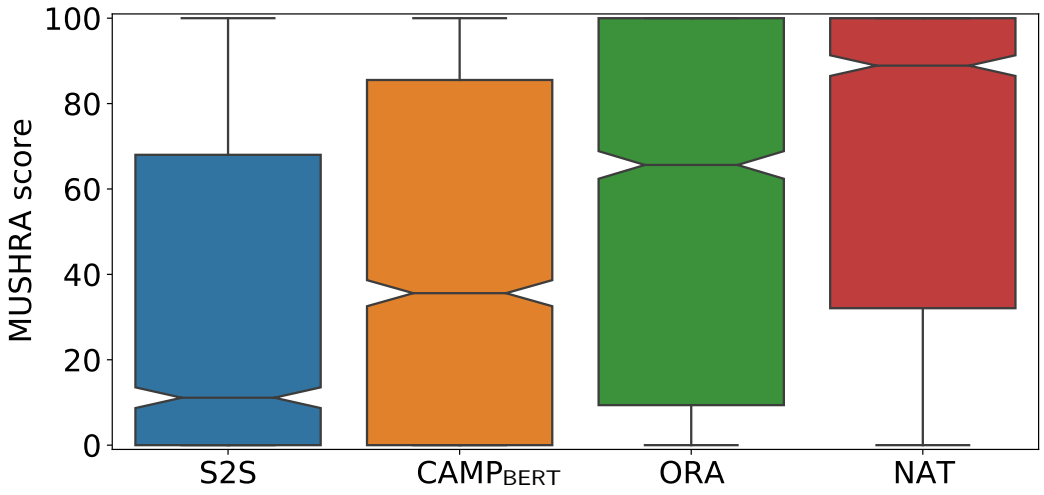



Figure C.2: Normalised MUSHRA listening test results with CAMP$_{\text{BERT}}$.

# Bibliography


Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*. [Cited in section 4.4.2.]

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S. (2016). YouTube-8M: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*. [Cited in section 7.1.]

Akuzawa, K., Iwasawa, Y., and Matsuo, Y. (2018). Expressive speech synthesis via modeling expressions with variational autoencoder. *arXiv preprint arXiv:1804.02135*. [Cited in section 2.2.5.]

Allbritton, D. W., McKoon, G., and Ratcliff, R. (1996). Reliability of prosodic cues for resolving syntactic ambiguity. *J. of experimental psychology. Learning, memory, and cognition*, 22(3):714–735. [Cited in section 6.3.2.2.]

An, X., Soong, F. K., Yang, S., and Xie, L. (2021). Effective and direct control of neural TTS prosody by removing interactions between different attributes. *Neural Networks*. [Cited in section 5.2.]

Anderson, A. H., Bader, M., Bard, E. G., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., et al. (1991). The HCRC map task corpus. *Language and speech*, 34(4):351–366. [Cited in section 2.2.2.]

Androutsopoulos, I. and Malakasiotis, P. (2010). A survey of paraphrasing and textual entailment methods. *J. of Artificial Intelligence Research*, 38:135–187. [Cited in section 1.]

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. (2020). Pseudo-labeling and confirmation bias in deep semi-supervised learning. In

*Proc. International Joint Conference on Neural Networks*, pages 1–8. IEEE. [Cited in section 4.3.]

Arik, S., Diamos, G., Gibiansky, A., Miller, J., Peng, K., Ping, W., Raiman, J., and Zhou, Y. (2017a). Deep voice 2: Multi-speaker neural text-to-speech. *arXiv preprint arXiv:1705.08947*. [Cited in section 2.2.7.]

Arik, S. O., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Raiman, J., Sengupta, S., et al. (2017b). Deep voice: Real-time neural text-to-speech. *arXiv preprint arXiv:1702.07825*. [Cited in section 2.2.7.]

Armstrong, M. E. and Prieto, P. (2015). The contribution of context and contour to perceived belief in polar questions. *J. of Pragmatics*, 81:77–92. [Cited in section 5.1.]

Arrow, K. J. (1950). A difficulty in the concept of social welfare. *J. of political economy*, 58(4):328–346. [Cited in section 7.]

Atal, B. S. and Hanauer, S. L. (1971). Speech analysis and synthesis by linear prediction of the speech wave. *J. of the Acoustical Society of America*, 50(2B):637–655. [Cited in section 2.1.2.]

Aubin, A., Cervone, A., Watts, O., and King, S. (2019). Improving speech synthesis with discourse relations. In *Proc. Interspeech*, pages 4470–4474. [Cited in sections 2.2.6.2 and 5.1.]

Austin, J. L. (1975). *How to do things with words*, volume 88. Oxford university press. [Cited in section 2.2.1.]

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., and Auli, M. (2022). Data2vec: A general framework for self-supervised learning in speech, vision and language. [Cited in section 2.2.5.]

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*. [Cited in sections 1.1.1, 2.2.5, 6.1, 6.5.4, and 7.1.]

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*. [Cited in sections 2.1.1.2, 2.5.1.4, and 6.4.2.]

Baljekar, P. and Black, A. W. (2016). Utterance selection techniques for TTS systems using found speech. In *Proc. Speech Synthesis Workshop*, pages 184–189, Sunnyvale, USA. [Cited in section 2.4.2.]

Banse, R. and Scherer, K. R. (1996). Acoustic profiles in vocal emotion expression. *Journal of personality and social psychology*, 70(3):614. [Cited in section 4.4.4.1.]

Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press. [Cited in section 2.5.2.]

Barra-Chicote, R., Yamagishi, J., King, S., Montero, J. M., and Macias-Guarasa, J. (2010). Analysis of statistical parametric and unit selection speech synthesis systems applied to emotional speech. *Speech Communication*, 52(5):394–404. [Cited in section 2.1.3.]

Bartels, C. (2014). *The intonation of English statements and questions: A compositional interpretation*. Routledge. [Cited in section 2.2.2.]

Barth-Weingarten, D., Dehé, N., and Wichmann, A. (2009). *Where prosody meets pragmatics*, volume 8. Brill. [Cited in section 2.2.1.]

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*. [Cited in section 7.1.]

Battenberg, E., Skerry-Ryan, R., Mariooryad, S., Stanton, D., Kao, D., Shannon, M., and Bagby, T. (2020). Location-relative attention mechanisms for robust long-form speech synthesis. In *Proc. International Conference on Speech and Signal Processing*, pages 6194–6198. IEEE. [Cited in section 2.1.1.2.]

Bawden, R., Clavel, C., and Landragin, F. (2016). Towards the generation of dialogue acts in socio-affective ecas: a corpus-based prosodic analysis. *Language Resources and Evaluation*, 50(4):821–838. [Cited in section 5.1.]

Ben-David, B. M., Multani, N., Shakuf, V., Rudzicz, F., and van Lieshout, P. H. (2016). Prosody and semantics are separate but not separable channels in the

perception of emotional speech: Test for rating of emotions in speech. *Journal of Speech, Language, and Hearing Research*, 59(1):72–89. [Cited in sections 1 and 2.2.]

Berthelot, D., Raffel, C., Roy, A., and Goodfellow, I. (2019). Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *Proc. International Conference on Learning Representations*. [Cited in section 5.3.2.]

Betz, S., Zarrieß, S., Székely, E., and Wagner, P. (2019). The greennn tree — lengthening position influences uncertainty perception. In *Proc. Interspeech*, pages 3990–3994. [Cited in section 5.1.]

Bezooijen, R. v. (1984). The characteristics and recognisability of vocal expression of emotions. [Cited in section 4.4.4.1.]

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451. [Cited in section 2.1.]

Bishop, C. M. (1994). Mixture density networks. Technical report, Citeseer. [Cited in section 3.3.]

Bishop, C. M. et al. (1995). *Neural networks for pattern recognition*. Oxford university press. [Cited in sections 2.5.1 and 2.5.1.6.]

Black, A., Taylor, P., Caley, R., and Clark, R. (1998). The festival speech synthesis system. [Cited in sections 2.1, 2.1, 2.2.7, and 2.5.]

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. [Cited in sections 2.5.1.1 and 7.1.]

Borg, I. and Groenen, P. (2003). Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement*, 40(3):277–280. [Cited in sections 2.3.2 and 3.5.4.1.]

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Driessche, G. v. d., Lespiau, J.-B., Damoc, B., Clark, A., et al. (2021). Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*. [Cited in section 7.1.]

Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs:

I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345. [Cited in section 3.6.]

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Proc. International Conference on Neural Information Processing Systems*, volume 33, pages 1877–1901. [Cited in section 7.1.]

Bryant, G. A. (2011). Verbal irony in the wild. *Pragmatics and Cognition*, 19(2):291–309. [Cited in section 2.2.]

BS Series (2014). Method for the subjective assessment of intermediate quality level of audio systems. *International Telecommunication Union Radiocommunication Assembly*. [Cited in sections 2.3.2, 3.5.2.1, 6.5.3.2, and 6.5.3.3.]

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335. [Cited in sections 4.1, 4.2.1, and 4.4.1.1.]

Byrne, D. (2021). Generative modelling for expressive speech. Master's thesis, University College London. [Cited in section 3.5.6.1.]

Cai, X., Dai, D., Wu, Z., Li, X., Li, J., and Meng, H. (2020). Emotion controllable speech synthesis using emotion-unlabeled dataset with the assistance of cross-domain speech emotion recognition. [Cited in sections 2.1.3 and 4.5.]

Calhoun, S. (2010). The centrality of metrical structure in signaling information structure: A probabilistic perspective. *Language*, 86(1):1–42. [Cited in sections 2.2 and 5.1.]

Campbell, N. (2007). Evaluation of speech synthesis. In *Evaluation of text and speech systems*, pages 29–64. Springer. [Cited in sections 1, 2.3, and 2.3.3.]

Canavan, Alexandra, D. G. and Zipperlen, G. (1997). CALLHOME amer-

ican english speech LDC97S42. [dataset]. Linguistic Data Consortium. doi.org/10.35111/exq3-x930. [Cited in section 2.4.2.]

Carletta, J. (2006). Announcing the AMI meeting corpus. *The ELRA Newsletter*, 11(1):3–5. [Cited in section 2.4.2.]

Caruana, R. (1998). Multitask learning. In *Learning to learn*, pages 95–133. Springer. [Cited in sections 1.1.2, 2.5.1.5, and 4.3.1.]

Cassell, J., Gill, A., and Tepper, P. (2007). Coordination in conversation and rapport. In *Workshop on Embodied Language Processing*, pages 41–50. [Cited in section 2.2.6.2.]

Cernak, M. and Rusko, M. (2005). An evaluation of synthetic speech using the PESQ measure. In *Proc. European Congress on Acoustics*, pages 2725–2728. [Cited in section 2.3.1.]

Chafe, W. L. (1976). Givenness, contrastiveness, definiteness, subjects, topics, and point of view. In Li, C. N., editor, *Subject and Topic*, pages 25–55. Academic Press, New York. [Cited in section 2.2.6.1.]

Chen, B., Bian, T., and Yu, K. (2017). Discrete duration model for speech synthesis. In *Proc. Interspeech*, pages 789–793, Stockholm, Sweden. [Cited in section 2.2.7.]

Chiu, C.-C. and Raffel, C. (2018). Monotonic chunkwise attention. In *Proc. International Conference on Learning Representations*. [Cited in section 2.1.1.2.]

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. [Cited in sections 2.5.1.2 and 3.5.1.]

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *Proc. International Conference on Neural Information Processing Systems*, pages 577–585, Montréal, Canada. [Cited in sections 2.1.1.2 and 6.4.1.]

Clark, H. H. and Brennan, S. E. (1991). Grounding in communication. [Cited in section 2.2.]

Clark, R., Silen, H., Kenter, T., and Leith, R. (2019). Evaluating long-form text-

to-speech: Comparing the ratings of sentences and paragraphs. In *Proc. Speech Synthesis Workshop*, pages 99–104, Vienna, Austria. [Cited in sections 1, 1.1.3, 2.3, 2.3.3, 2.4.1, 6.1, 6.5.3.2, 6.5.3.3, and 7.1.]

Clark, R. A. J., Podsiadlo, M., Fraser, M., Mayo, C., and King, S. (2007). Statistical analysis of the Blizzard challenge 2007 listening test results. Bonn, Germany. [Cited in section 3.5.3.]

Clifton, A., Reddy, S., Yu, Y., Pappu, A., Rezapour, R., Bonab, H., Eskevich, M., Jones, G., Karlgren, J., Carterette, B., and Jones, R. (2020). 100,000 podcasts: A spoken English document corpus. In *Proc. International Conference on Computational Linguistics*, pages 5903–5917, Barcelona, Spain. [Cited in sections 2.4.2 and 7.1.]

Clynes, M. (1977). Sentics: The touch of the emotions. *Anchor Press/Doubleday*. [Cited in sections 1 and 2.2.6.2.]

Cole, J. (2015). Prosody in context: a review. *Language, Cognition and Neuroscience*, 30(1-2):1–31. [Cited in sections 1, 2.2.1, 2.2.2, 2.2.6, and 2.2.6.1.]

Cole, J., Mahrt, T., and Hualde, J. I. (2014). Listening for sound, listening for meaning: Task effects on prosodic transcription. In *Proc. Speech Prosody*, pages 859–863. [Cited in section 2.3.3.]

Cole, J., Mahrt, T., and Roy, J. (2017). Crowd-sourcing prosodic annotation. *Computer Speech and Language*, 45:300–325. [Cited in sections 2.2, 2.2.2, 2.2.3, 2.2.4, 4.1, 5.1, and 5.3.1.]

Cole, J. and Reichel, U. (2016). What entrainment reveals about the cognitive encoding of prosody and its relation to discourse function. In *Proc. Speech Prosody*. [Cited in section 5.1.]

Constant, N. (2012). English rise-fall-rise: a study in the semantics and pragmatics of intonation. *Linguistics and Philosophy*, 35(5):407–442. [Cited in section 2.2.2.]

Cortina, J. M. (1993). What is coefficient alpha? an examination of theory and applications. [Cited in section 4.2.1.]

Cotescu, M., Drugman, T., Huybrechts, G., Lorenzo-Trueba, J., and Moinet, A.

(2019). Voice conversion for whispered speech synthesis. *IEEE Signal Processing Letters*, 27:186–190. [Cited in section 2.4.]

Coupé, C., Oh, Y. M., Dediu, D., and Pellegrino, F. (2019). Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche. *Science Advances*, 5(9). [Cited in sections 2.1 and 2.]

Cowie, R. and Cornelius, R. R. (2003). Describing the emotional states that are expressed in speech. *Speech Communication*, 40(1):5–32. [Cited in section 4.2.1.]

Csáji, B. C. et al. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24(48):7. [Cited in section 2.5.1.]

Cutler, A., Dahan, D., and Van Donselaar, W. (1997). Prosody in the comprehension of spoken language: A literature review. *Language and speech*, 40(2):141–201. [Cited in section 2.2.1.]

Dai, B. and Wipf, D. (2019). Diagnosing and enhancing VAE models. In *Proc. International Conference on Learning Representations*, New Orleans, USA. [Cited in sections 3.4 and 3.]

Dall, R., Hashimoto, K., Oura, K., Nankaku, Y., and Tokuda, K. (2016). Redefining the linguistic context feature set for HMM and DNN TTS through position and parsing. In *Proc. Interspeech*, pages 2851–2855, San Francisco, USA. [Cited in sections 1.1.3, 2.2.6.1, and 2.2.6.2.]

Dall, R., Tomalin, M., Wester, M., Byrne, W., and King, S. (2014). Investigating automatic and human filled pause insertion for speech synthesis. In *Proc. Interspeech*. [Cited in section 2.2.7.]

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap methods and their application*. Number 1. Cambridge university press. [Cited in section 6.5.3.2.]

De Moraes, J. A. (2011). From a prosodic point of view: remarks on attitudinal meaning. pages 19–37. [Cited in sections 1.1.1, 2.2.1, and 2.2.6.1.]

Deese, J. and Kaufman, R. A. (1957). Serial effects in recall of unorganized and sequentially organized verbal material. *Journal of experimental psychology*, 54(3):180. [Cited in section 2.3.3.]

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. (2011). Front-

end factor analysis for speaker verification. *IEEE Trans. on Audio, Speech, and Language Processing*, 19(4):788–798. [Cited in section 2.1.3.]

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*, pages 4171–4186. [Cited in sections 2.2.5, 2.5.1, and 6.3.2.3.]

Dilley, L. and Brown, M. (2005). The RaP (rhythm and pitch) labeling system. [Cited in section 2.2.]

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real MVP. In *Proc. International Conference on Learning Representations*, Toulon, France. [Cited in section 2.5.2.]

Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*. [Cited in section 2.5.2.1.]

Domingos, P. (2020). Every model learned by gradient descent is approximately a kernel machine. *arXiv preprint arXiv:2012.00152*. [Cited in sections 2.4 and 2.5.1.]

Dorta, G., Vicente, S., Agapito, L., Campbell, N. D. F., and Simpson, I. (2018). Structured uncertainty prediction networks. In *Proc. CVPR*. [Cited in section 2.5.2.1.]

Douglas-Cowie, E., Campbell, N., Cowie, R., and Roach, P. (2003). Emotional speech: Towards a new generation of databases. *Speech Communication*, 40(1):33–60. [Cited in sections 2.2.6.2, 2.4.2, 4.1, 4.2.1, 1, 4.3.1, and 4.4.1.1.]

Drugman, T., Huybrechts, G., Klimkov, V., and Moinet, A. (2018). Traditional machine learning for pitch detection. *IEEE Signal Processing Letters*, 25(11):1745–1749. [Cited in section 2.2.3.]

Dunbar, E., Algayres, R., Karadayi, J., Bernard, M., Benjumea, J., Cao, X.-N., Miskic, L., Dugrain, C., Ondel, L., Black, A. W., et al. (2019). The zero resource speech challenge 2019: TTS without T. In *Proc. Interspeech*, pages 1088–1092, Graz, Austria. [Cited in section 6.2.]

Ebden, P. and Sproat, R. (2015). The Kestrel TTS text normalization system. *Natural Language Engineering*, 21(3):333–353. [Cited in section 2.1.]

Ekman, P. (1992). An argument for basic emotions. *Cognition and emotion*, 6(3-4):169–200. [Cited in sections 1.1.1 and 4.2.1.]

Ekman, P., Friesen, W. V., O'sullivan, M., Chan, A., Diacoyanni-Tarlatzis, I., Heider, K., Krause, R., LeCompte, W. A., Pitcairn, T., Ricci-Bitti, P. E., et al. (1987). Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of personality and social psychology*, 53(4):712. [Cited in section 4.2.1.]

Elias, I., Zen, H., Shen, J., Zhang, Y., Jia, Y., Skerry-Ryan, R., and Wu, Y. (2021). Parallel Tacotron 2: A non-autoregressive neural TTS model with differentiable duration modeling. *arXiv preprint arXiv:2103.14574*. [Cited in sections 1.1.3, 2.3, 2.3.2, and 6.1.]

Elias, I., Zen, H., Shen, J., Zhang, Y., Jia, Y., Weiss, R., and Wu, Y. (2020). Parallel Tacotron: Non-autoregressive and controllable TTS. *arXiv preprint arXiv:2010.11439*. [Cited in section 2.1.1.2.]

Erhan, D., Bengio, Y., Courville, A., and Vincent, P. (2009). Visualizing higher-layer features of a deep network. Technical Report Tech. rep. 1341, University of Montreal. [Cited in section 2.5.1.3.]

Espic, F., Valentini-Botinhao, C., and King, S. (2017). Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis. In *Proc. Interspeech*, volume 5, Stockholm, Sweden. [Cited in section 2.1.2.]

Eyben, F., Scherer, K. R., Schuller, B. W., Sundberg, J., André, E., Busso, C., Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S., et al. (2016). The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. *IEEE Trans. on Affective Computing*, 7(2):190–202. [Cited in sections 4.2.2, 4.3.1, and 4.1.]

Fang, W., Chung, Y.-A., and Glass, J. (2019). Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models. *arXiv preprint arXiv:1906.07307*. [Cited in sections 2.2.6.1, 2.2.6.2, 6.1, and 6.5.3.3.]

Farrús, M., Lai, C., and Moore, J. D. (2016). Paragraph-based prosodic cues for speech synthesis applications. *Proc. Speech Prosody*. [Cited in section 2.2.4.]

Farrús, M., Lai, C., and Moore, J. D. (2016). Paragraph-based prosodic cues for speech synthesis applications. In *Proc. Speech Prosody*, pages 1143–1147, Boston, MA, USA. [Cited in section 5.1.]

Fernandez, R., Rendel, A., Ramabhadran, B., and Hoory, R. (2014). Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks. In *Proc. Interspeech*, pages 2268–2272, Singapore. [Cited in sections 1.1.1 and 2.2.7.]

Fisher, R. (1953). Dispersion on a sphere. *Proc. R. Soc. London A: Mathematical, Physical and Engineering Sciences*, 217(1130):295–305. [Cited in section 3.4.]

Fitt, S. and Isard, S. (1999). Synthesis of regional English using a keyword lexicon. In *Proc. Eurospeech*, pages 823–826. [Cited in sections 2.1 and 3.5.1.]

Flipsen Jr, P. (2006). Syllables per word in typical and delayed speech acquisition. *Clinical linguistics and phonetics*, 20(4):293–301. [Cited in section 6.3.1.]

Fong, J., Gallegos, P. O., Hodari, Z., and King, S. (2019). Investigating the robustness of sequence-to-sequence text-to-speech models to imperfectly-transcribed training data. In *Proc. Interspeech*, pages 1546–1550. [Cited in section 1.2.1.]

Fontaine, J. R., Scherer, K. R., Roesch, E. B., and Ellsworth, P. C. (2007). The world of emotions is not two-dimensional. *Psychological science*, 18(12):1050–1057. [Cited in sections 1.1.1 and 4.2.1.]

Freeman, V. (2019). Prosodic features of stances in conversation. *J. of the Association for Laboratory Phonology*, 10(1). [Cited in section 5.1.]

Gallegos, P. O., Williams, J., Rownicka, J., and King, S. (2020). An unsupervised method to select a speaker subset from large multi-speaker speech synthesis datasets. In *Proc. Interspeech*, pages 1758–1762, Shanghai, China. [Cited in section 2.4.2.]

Gatt, A. and Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. of Artificial Intelligence Research*, 61:65–170. [Cited in section 1.]

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58. [Cited in section 2.4.1.]

Ghosh, S., Laksana, E., Morency, L.-P., and Scherer, S. (2016). Representation learning for speech emotion recognition. In *INTERSPEECH*, pages 3603–3607. [Cited in section 4.2.2.]

Gideon, J., Khorram, S., Aldeneh, Z., Dimitriadis, D., and Provost, E. M. (2017). Progressive neural networks for transfer learning in emotion recognition. *arXiv preprint arXiv:1706.03256*. [Cited in section 4.1.]

Gironzetti, E. (2017). Prosodic and multimodal markers of humor. In *The Routledge handbook of language and humor*, pages 400–413. Routledge. [Cited in section 2.2.]

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proc. Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings. [Cited in section 2.5.1.]

Gobl, C. and Chasaide, A. N. (2003). The role of voice quality in communicating emotion, mood and attitude. *Speech Communication*, 40(1-2):189–212. [Cited in sections 2.2.1 and 2.2.3.]

Godfrey, J. J., Holliman, E. C., and McDaniel, J. (1992). SWITCHBOARD: Telephone speech corpus for research and development. In *Proc. International Conference on Speech and Signal Processing*, volume 1, pages 517–520. IEEE. [Cited in section 2.2.2.]

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. [Cited in sections 2.5.1 and 2.5.1.5.]

Goodhue, D., Harrison, L., Su, Y. C., and Wagner, M. (2016). Toward a bestiary of English intonational contours. In *Proc. North East Linguistics Society*, pages 311–320. [Cited in sections 1, 1.1.3, 2.2, 2.2.2, 2.3.3, 2.4.2, 3.1, 4.1, 5.4.2.2, and 6.1.]

Goodhue, D. and Wagner, M. (2018). Intonation, yes and no. *Glossa: a journal of general linguistics*, 3(1). [Cited in section 2.2.2.]

Goodwin, C., Duranti, A., Hanks, W. F., Duranti, A., Lindstrom, L., Bauman, R., Goodwin, C., Goodwin, M. H., Schegloff, E. A., Gumperz, J., Basso, E., Gaik, F., Cicourel, A., Philips, S. U., Kendon, A., and Ochs, E. (1992). *Rethinking context: Language as an interactive phenomenon*. Cambridge University Press. [Cited in sections 2.2.6 and 2.2.6.1.]

Govender, A. and King, S. (2018). Using pupillometry to measure the cogni-

tive load of synthetic speech. In *Proc. Interspeech*, pages 2838–2842. [Cited in section 2.3.]

Gravano, A., Benus, S., Hirschberg, J., German, E. S., and Ward, G. (2008a). The effect of contour type and epistemic modality on the assessment of speaker certainty. In *Proc. Speech Prosody*, pages 401–404. [Cited in section 5.1.]

Gravano, A., Benus, S., Hirschberg, J. B., German, E. S., and Ward, G. (2008b). The effect of contour type and epistemic modality on the assessment of speaker certainty. [Cited in section 2.2.2.]

Gravano, A. and Hirschberg, J. (2009). Turn-yielding cues in task-oriented dialogue. In *Proc. SIGDIAL*, pages 253–261. [Cited in section 2.2.]

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*. [Cited in section 2.1.1.2.]

Grice, M., Ritter, S., Niemann, H., and Roettger, T. B. (2017). Integrating the discreteness and continuity of intonational categories. *J. of Phonetics*, 64:90–107. [Cited in section 5.1.]

Grice, P. (1989). *Studies in the Way of Words*. Harvard University Press. [Cited in section 2.2.6.]

Gu, J., Lu, Z., Li, H., and Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 1631–1640, Berlin, Germany. [Cited in section 2.1.1.2.]

Gutierrez, E., Oplustil-Gallegos, P., and Lai, C. (2021). Location, location: Enhancing the evaluation of text-to-speech synthesis using the rapid prosody transcription paradigm. In *Proc. Speech Synthesis Workshop*, pages 25–30. [Cited in section 7.1.]

Habib, R., Mariooryad, S., Shannon, M., Battenberg, E., Skerry-Ryan, R., Stanton, D., Kao, D., and Bagby, T. (2020). Semi-supervised generative modeling for controllable speech synthesis. In *International Conference on Learning Representations*. [Cited in section 2.2.3.]

Halliday, M. A. K. and Matthiessen, C. (1999). *Construing experience through*

*meaning: A language-based approach to cognition.* Bloomsbury Publishing. [Cited in sections 2.2.6.1 and 6.3.2.3.]

Hamon, C., Mouline, E., and Charpentier, F. (1989). A diphone synthesis system based on time-domain prosodic modifications of speech. In *Proc. International Conference on Speech and Signal Processing*, pages 238–241, Glasgow, UK. [Cited in section 2.1.]

Hayashi, T., Watanabe, S., Toda, T., Takeda, K., Toshniwal, S., and Livescu, K. (2019). Pre-trained text embeddings for enhanced text-to-speech synthesis. In *Proc. Interspeech*, pages 4430–4434, Graz, Austria. [Cited in sections 2.2.6.1, 2.2.6.2, 6.1, and 6.5.3.3.]

He, M., Deng, Y., and He, L. (2019). Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural TTS. In *Proc. Interspeech*, pages 1293–1297. [Cited in sections 2.1.1.2 and 6.4.2.]

Henter, G. E., Lorenzo-Trueba, J., Wang, X., Kondo, M., and Yamagishi, J. (2018a). Cyborg speech: Deep multilingual speech synthesis for generating segmental foreign accent with natural prosody. In *Proc. International Conference on Speech and Signal Processing*, pages 4799–4803, Calgary, Canada. IEEE. [Cited in section 2.1.3.]

Henter, G. E., Lorenzo-Trueba, J., Wang, X., and Yamagishi, J. (2017a). Principles for learning controllable TTS from annotated and latent variation. In *Proc. Interspeech*, volume 2017, pages 3956–3960, Stockholm, Sweden. [Cited in section 2.1.3.]

Henter, G. E., Lorenzo-Trueba, J., Wang, X., and Yamagishi, J. (2018b). Deep encoder-decoder models for unsupervised learning of controllable speech synthesis. *arXiv preprint arXiv:1807.11470*. [Cited in sections 2.1.3, 3.2, and 5.2.]

Henter, G. E., Merritt, T., Shannon, M., Mayo, C., and King, S. (2014). Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech. In *Proc. Interspeech*, pages 1504–1508, Singapore. [Cited in section 3.3.]

Henter, G. E., Ronanki, S., Watts, O., and King, S. (2017b). Non-parametric duration modelling for speech synthesis with a joint model of acoustics and

duration. *IEICE technical report*, 116(414):11–16. [Cited in sections 1.1.1, 2.2.7, and 5.4.1.2.]

Henter, G. E., Ronanki, S., Watts, O., Wester, M., Wu, Z., and King, S. (2016). Robust TTS duration modelling using DNNs. In *Proc. International Conference on Speech and Signal Processing*, pages 5130–5134, Shanghai, China. [Cited in section 2.2.7.]

Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., and Esteve, Y. (2018). Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation. *arXiv preprint arXiv:1805.04699*. [Cited in section 2.4.2.]

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). $\beta$-VAE: Learning basic visual concepts with a constrained variational framework. In *Proc. International Conference on Learning Representations*, volume 3, Toulon, France. [Cited in section 2.5.2.1.]

Hinterleitner, F., Neitzel, G., Möller, S., and Norrenbrock, C. (2011). An evaluation protocol for the subjective assessment of text-to-speech in audiobook reading tasks. [Cited in section 2.3.3.]

Hirschberg, J. (1999). Communication and prosody: Functional aspects of prosody. *Speech Communication*, 36(1-2):31–43. [Cited in section 2.2.2.]

Hirschberg, J. and Pierrehumbert, J. (1986). The intonational structuring of discourse. In *24th Annual Meeting of the Association for Computational Linguistics*, pages 136–144. [Cited in section 2.2.1.]

Hirschberg, J. and Prieto, P. (1996). Training intonational phrasing rules automatically for English and Spanish text-to-speech. *Speech Communication*, 18(3):283–292. [Cited in section 2.2.7.]

Hirschberg, J. and Ward, G. (1992). The influence of pitch range, duration, amplitude and spectral features on the interpretation of the rise-fall-rise intonation contour in English. *J. of phonetics*, 20(2):241–251. [Cited in section 2.2.2.]

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851. [Cited in section 2.5.2.]

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. [Cited in section 2.5.1.2.]

Hodari, Z. (2017a). A learned emotion space for emotion recognition and emotive speech synthesis. Master's thesis, The University of Edinburgh. [Cited in sections 1.2.1, 4.3.1, and 4.1.]

Hodari, Z. (2017b). modNN: A modular TensorFlow interface. *github.com/ZackHodari/modNN*. [Cited in section 4.4.2.]

Hodari, Z. (2020a). Morgana: A toolkit for defining and training TTS voices in PyTorch. *github.com/ZackHodari/morgana*. [Cited in sections 1.2.1, 3.5.1, and 5.4.1.]

Hodari, Z. (2020b). tts-data-tools: Data processing tools for preparing speech and text for training TTS voices. *github.com/ZackHodari/tts_data_tools*. [Cited in sections 1.2.1 and 5.4.1.]

Hodari, Z., Lai, C., and King, S. (2020). Perception of prosodic variation for speech synthesis using an unsupervised discrete representation of F0. In *Proc. Speech Prosody*, pages 965–969, Tokyo, Japan. [Cited in sections 1.2 and 5.]

Hodari, Z., Moinet, A., Karlapati, S., Lorenzo-Trueba, J., Merritt, T., Joly, A., Abbas, A., Karanasou, P., and Drugman, T. (2021). CAMP: A two-stage approach to modelling prosody in context. In *Proc. International Conference on Speech and Signal Processing*, Toronto, Canada. [Cited in sections 1.2 and 6.]

Hodari, Z., Watts, O., and King, S. (2019). Using generative modelling to produce varied intonation for speech synthesis. In *Proc. Speech Synthesis Workshop*, pages 239–244, Vienna, Austria. [Cited in sections 1.2 and 3.]

Hodari, Z., Watts, O., Ronanki, S., and King, S. (2018). Learning interpretable control dimensions for speech synthesis by using external data. In *Proc. Interspeech*, pages 32–36, Hyderabad, India. [Cited in sections 1.2 and 4.]

Hoffman, M. D. and Johnson, M. J. (2016). ELBO surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NeurIPS*, volume 1, page 2. [Cited in section 3.4.]

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70. [Cited in section 2.3.2.]

Holschneider, M., Kronland-Martinet, R., Morlet, J., and Tchamitchian, P.

(1990). A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297, Berlin, Heidelberg. Springer. [Cited in section 2.5.1.3.]

Hsu, C.-C., Hwang, H.-T., Wu, Y.-C., Tsao, Y., and Wang, H.-M. (2017a). Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks. *arXiv preprint arXiv:1704.00849*. [Cited in section 2.2.5.]

Hsu, W.-N., Zhang, Y., and Glass, J. (2017b). Unsupervised learning of disentangled and interpretable representations from sequential data. In *Proc. International Conference on Neural Information Processing Systems*, pages 1878–1889, Long Beach, USA. [Cited in section 1.1.2.]

Hsu, W.-N., Zhang, Y., Weiss, R., Zen, H., Wu, Y., Cao, Y., and Wang, Y. (2019). Hierarchical generative modeling for controllable speech synthesis. In *Proc. International Conference on Learning Representations*, New Orleans, USA. [Cited in sections 2.1 and 2.2.5.]

Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. (2018). Multimodal unsupervised image-to-image translation. In *Proc. European conference on computer vision*, pages 172–189. [Cited in section 7.1.]

Hübscher, I., Garufi, M., and Prieto, P. (2018). Preschoolers use prosodic mitigation strategies to encode polite stance. In *Proc. Speech Prosody*, pages 255–259. [Cited in section 5.1.]

Hunt, A. J. and Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *Proc. International Conference on Speech and Signal Processing*, pages 373–376, Atlanta, USA. [Cited in section 2.1.]

Ijima, Y., Hojo, N., Masumura, R., and Asami, T. (2017). Prosody aware word-level encoder based on BLSTM-RNNs for DNN-based speech synthesis. In *Proc. Interspeech*, pages 764–768, Stockholm, Sweden. [Cited in section 2.2.6.1.]

Imai, S. (1983). Cepstral analysis synthesis on the mel frequency scale. In *Proc. International Conference on Speech and Signal Processing*, volume 8, pages 93–96. [Cited in section 2.1.]

Itô, J. (2018). *Syllable theory in prosodic phonology*, volume 10. Routledge. [Cited in section 2.2.4.]

Ito, K. (2017). The LJ speech dataset. [dataset]. keithito.com/LJ-Speech-Dataset. [Cited in section 2.4.2.]

Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Proc. International Conference on Neural Information Processing Systems*, pages 8580–8589, Montréal, Canada. [Cited in sections 2.4 and 2.5.1.]

Janssen, J. (1957). A method for the calculation of the speech intelligibility under conditions of reverberation and noise. *Acta Acustica united with Acustica*, 7(5):305–310. [Cited in section 2.3.1.]

Jia, Y., Zen, H., Shen, J., Zhang, Y., and Wu, Y. (2021). PnG BERT: Augmented BERT on phonemes and graphemes for neural TTS. In *Proc. Interspeech*, pages 151–155. [Cited in sections 2.3 and 2.3.2.]

Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., Nguyen, P., Pang, R., Lopez Moreno, I., and Wu, Y. (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Proc. International Conference on Neural Information Processing Systems*, 31. [Cited in section 2.1.3.]

Jin, Z., Finkelstein, A., Mysore, G. J., and Lu, J. (2018). FFTNet: A real-time speaker-dependent neural vocoder. In *Proc. International Conference on Speech and Signal Processing*, pages 2251–2255. [Cited in section 2.1.2.]

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233. [Cited in section 2.5.2.]

Juvela, L., Bollepalli, B., Yamagishi, J., and Alku, P. (2019). GELP: GAN-excited linear prediction for speech synthesis from mel-spectrogram. In *Proc. Interspeech*, pages 694–698, Graz, Austria. [Cited in section 2.1.2.]

Kahn, J., Riviere, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P.-E., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C., et al. (2020). Libri-Light: A benchmark for ASR with limited or no supervision. In *Proc. International*

*Conference on Speech and Signal Processing*, pages 7669–7673. [Cited in section 7.1.]

Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., van den Oord, A., Dieleman, S., and Kavukcuoglu, K. (2018). Efficient neural audio synthesis. *arXiv preprint arXiv:1802.08435*. [Cited in sections 1.1.3, 2.1.2, and 2.3.]

Kane, J. and Gobl, C. (2011). Identifying regions of non-modal phonation using features of the wavelet transform. In *Proc. Interspeech*, Florence, Italy. [Cited in section 2.2.3.]

Kang, M., Lee, J., Kim, S., and Kim, I. (2021). Fast DCTTS: Efficient deep convolutional text-to-speech. *arXiv preprint arXiv:2104.00624*. [Cited in sections 2.1.3 and 2.5.1.3.]

Karbasi, M. and Kolossa, D. (2017). ASR-based measures for microscopic speech intelligibility prediction. pages 67–70. [Cited in section 2.3.1.]

Karlapati, S., Abbas, A., Hodari, Z., Moinet, A., Joly, A., Karanasou, P., and Drugman, T. (2021). Prosodic representation learning and contextual sampling for neural text-to-speech. In *Proc. International Conference on Speech and Signal Processing*, Toronto, Canada. [Cited in sections 1.1.3, 1.2.1, 2.2.5, 2.2.6.2, 5.2, 6.2, 1, and 7.1.]

Karlapati, S., Moinet, A., Joly, A., Klimkov, V., Sáez-Trigueros, D., and Drugman, T. (in *Proc. Interspeech*, 2020). CopyCat: Many-to-many fine-grained prosody transfer for neural text-to-speech. [Cited in section 6.3.1.]

Kawahara, H. (2006). STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds. *Acoustical science and technology*, 27(6):349–353. [Cited in sections 2.1, 2.1.1.1, 2.1.2, and 4.3.2.]

Kenter, T., Sharma, M. K., and Clark, R. (2020). Improving prosody of RNN-based English text-to-speech synthesis by incorporating a BERT model. [Cited in sections 2.2.6.1 and 2.2.6.2.]

Kim, J. W., Salamon, J., Li, P., and Bello, J. P. (2018). CREPE: A convolu-

tional representation for pitch estimation. In *Proc. International Conference on Speech and Signal Processing*, pages 161–165. [Cited in section 2.2.3.]

Kim, Y., Lee, H., and Provost, E. M. (2013). Deep learning for robust feature generation in audiovisual emotion recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3687–3691. IEEE. [Cited in section 4.2.1.]

King, S., Black, A. W., Tokuda, K., and Prahallad, K. (2013). The Blizzard challenge 2013. Barcelona, Catalonia. [Cited in section 2.4.2.]

King, S., Crumlish, J., Martin, A., and Wihlborg, L. (2018). The Blizzard challenge 2018. Hyderabad, India. [Cited in section 4.1.]

King, S. and Karaiskos, V. (2016). The Blizzard challenge 2016. Cupertino, California. [Cited in sections 2.4.2 and 2.4.3.]

King, S., Wihlborg, L., and Guo, W. (2017). The Blizzard challenge 2017. Stockholm, Sweden. [Cited in section 4.4.4.1.]

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. [Cited in sections 2.5.1, 3.5.1, 5.4.1, and 6.5.2.]

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*. [Cited in sections 2.2.5, 2.5.2, 2.5.2.1, 2.5.2.1, 2.5.2.1, 3.1, 3.4, and 5.3.2.]

Kirk, R. E. (2013). *Experimental design: procedures for the behavioral sciences*. SAGE Publications, fourth edition. edition. [Cited in section 2.3.2.]

Klabbers, E., Mishra, T., and van Santen, J. P. (2007). Analysis of affective speech recordings using the superpositional intonation model. In *Proc. Speech Synthesis Workshop*, pages 339–344, Bonn, Germany. [Cited in sections 2.2.6.1 and 4.4.4.1.]

Kleinhans, J., Farrús, M., Gravano, A., Pérez, J. M., Lai, C., and Wanner, L. (2017). Using prosody to classify discourse relations. In *Proc. Interspeech*, pages 3201–3205. [Cited in section 5.1.]

Klimkov, V., Nadolski, A., Moinet, A., Putrycz, B., Barra-Chicote, R., Merritt, T., and Drugman, T. (2017). Phrase break prediction for long-form reading

TTS: exploiting text structure information. In *Proc. Interspeech*, pages 1064–1068, Stockholm, Sweden. [Cited in sections 2.1.3, 2.2.7, and 2.3.3.]

Klimkov, V., Ronanki, S., Rohnke, J., and Drugman, T. (2019). Fine-grained robust prosody transfer for single-speaker neural text-to-speech. In *Proc. Interspeech*, pages 4440–4444, Graz, Austria. [Cited in sections 1.1.1, 2.1.3, 5.2, and 6.2.]

Kochanski, G., Grabe, E., Coleman, J., and Rosner, B. (2005). Loudness predicts prominence: Fundamental frequency lends little. *J. of the Acoustical Society of America*, 118(2):1038–1054. [Cited in section 2.2.3.]

Köhn, A., Baumann, T., and Dörfler, O. (2018). An empirical analysis of the correlation of syntax and prosody. In *Proc. Interspeech*, pages 2157–2161. [Cited in sections 2.2.5, 5.3.1, 6.1, and 6.3.2.2.]

Kominek, J. and Black, A. (2006). The Blizzard challenge 2006 CMU entry introducing hybrid trajectory-selection synthesis. [Cited in section 2.1.]

Kominek, J. and Black, A. W. (2004). The CMU arctic speech databases. In *Proc. Speech Synthesis Workshop*. [Cited in section 1.2.1.]

Koutnik, J., Greff, K., Gomez, F., and Schmidhuber, J. (2014). A clockwork RNN. In *Proc. International Conference on Machine Learning*, pages 1863–1871. [Cited in section 2.5.1.2.]

Kraft, S. and Zölzer, U. (2014). BeaqleJS: HTML5 and JavaScript based framework for the subjective evaluation of audio quality. In *Linux Audio Conference, Karlsruhe, DE*. [Cited in section 4.4.4.]

Krifka, M. (2008). Basic notions of information structure. *Acta Linguistica Hungarica*, 55(3-4):243–276. [Cited in sections 1, 2.2.6, 2.2.6.1, and 6.3.2.3.]

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Proc. International Conference on Neural Information Processing Systems*, 25. [Cited in section 2.5.1.]

Ladd, D. R. (1980). *The structure of intonational meaning: Evidence from English*. Indiana University Press. [Cited in section 2.2.2.]

Ladd, D. R. (1986). Intonational phrasing: The case for recursive prosodic structure. *Phonology Yearbook*, 3:311–340. [Cited in section 2.2.4.]

Ladd, D. R. (2008). *Intonational phonology*. Cambridge University Press. [Cited in sections 2.2.2, 5.3.1, 5.3.2, and 6.1.]

Lai, C. (2010). What do you mean, you're uncertain?: the interpretation of cue words and rising intonation in dialogue. In *Proc. Interspeech*, pages 1413–1416. [Cited in section 5.1.]

Lai, C. (2012a). Response types and the prosody of declaratives. In *Proc. Speech Prosody*. [Cited in sections 2.2.2 and 5.1.]

Lai, C. (2012b). *Rises all the way up: The interpretation of prosody, discourse attitudes and dialogue structure*. PhD thesis, University of Pennsylvania. [Cited in section 2.2.6.1.]

Łańcucki, A. (2021). FastPitch: Parallel text-to-speech with pitch prediction. In *Proc. International Conference on Speech and Signal Processing*, pages 6588–6592. IEEE. [Cited in sections 2.1.1.1 and 2.1.1.2.]

Latorre, J. and Akamine, M. (2008). Multilevel parametric-base F0 model for speech synthesis. In *Proc. Interspeech*. [Cited in section 2.2.7.]

Latorre, J., Yanagisawa, K., Wan, V., Kolluru, B., and Gales, M. J. (2014). Speech intonation for TTS: Study on evaluation methodology. In *Proc. Interspeech*, pages 2957–2961, Singapore. [Cited in sections 1.1.3, 2.3, 2.3.3, and 3.5.2.]

Lawrence, C., Kotnis, B., and Niepert, M. (2019). Attending to future tokens for bidirectional sequence generation. In *Proc. Empirical Methods in Natural Language Processing*, pages 1–10, Hong Kong, China. [Cited in section 6.5.4.]

Lazarus, R. S. (1991). *Emotion and adaptation*. Oxford University Press. [Cited in section 4.2.1.]

Le-Khac, P. H., Healy, G., and Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934. [Cited in section 7.1.]

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324. [Cited in section 2.5.1.3.]

Lee, C.-C., Mower, E., Busso, C., Lee, S., and Narayanan, S. (2011). Emotion

recognition using a hierarchical binary decision tree approach. *Speech Communication*, 53(9):1162–1171. [Cited in section 4.2.1.]

Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3. [Cited in sections 1.1.2, 4.1, and 4.3.]

Lee, J. and Tashev, I. (2015). High-level feature representation using recurrent neural network for speech emotion recognition. In *Proc. Interspeech*, pages 1537–1540, Dresden, Germany. [Cited in sections 4.2.1, 4.4.2, and 4.1.]

Lei, M., Wu, Y., Soong, F. K., Ling, Z.-H., and Dai, L. (2010). A hierarchical F0 modeling method for HMM-based speech synthesis. In *Proc. Interspeech*, Chiba, Japan. [Cited in section 2.2.7.]

Lewis, D. (1979). Scorekeeping in a language game. In *Semantics from different points of view*, pages 172–187. Springer. [Cited in sections 1, 2.2.6.1, 2.2.6.2, and 6.3.2.3.]

Li, N., Liu, S., Liu, Y., Zhao, S., and Liu, M. (2019). Neural speech synthesis with Transformer network. In *Proc. AAAI Conference on Artificial Intelligence*, pages 6706–6713. [Cited in sections 2.1.1.2 and 2.5.1.]

Liberman, M. and Sag, I. (1974). Prosodic form and discourse function. In *Proc. of Chicago Linguistics Society*, volume 10, pages 402–415. [Cited in section 2.2.2.]

Liberman, M. Y. and Church, K. W. (1992). Text analysis and word pronunciation in text-to-speech synthesis. *Furui and Sondhi, Advances in Speech Technology*, pages 791–832. [Cited in section 5.3.1.]

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*. [Cited in section 2.3.2.]

Lim, D., Jang, W., Park, H., Kim, B., Yoon, J., et al. (2020). JDI-T: Jointly trained duration informed transformer for text-to-speech without explicit alignment. *arXiv preprint arXiv:2005.07799*. [Cited in sections 2.2.7 and 2.6.]

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Trans. on information theory*, 28(2):129–137. [Cited in section 2.]

Lo, C.-C., Fu, S.-W., Huang, W.-C., Wang, X., Yamagishi, J., Tsao, Y., and Wang, H.-M. (2019). MOSNet: Deep learning-based objective assessment for

voice conversion. In *Proc. Interspeech*, pages 1541–1545, Graz, Austria. [Cited in sections 2.3.1 and 3.5.6.1.]

Lorenzo-Trueba, J., Yamagishi, J., Toda, T., Saito, D., Villavicencio, F., Kinnunen, T., and Ling, Z. (2018). The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods. *arXiv preprint arXiv:1804.04262*. [Cited in section 1.2.1.]

Lotfian, R. and Busso, C. (2017). Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings. *IEEE Trans. on Affective Computing*, 10(4):471–483. [Cited in section 2.4.2.]

Loupi, D. (2017). An experimental assessment of the effects and limitations in adding context to the evaluation of prosody. Master's thesis, University of Edinburgh. [Cited in sections 1.1.3 and 2.3.]

Luong, H.-T., Takaki, S., Henter, G. E., and Yamagishi, J. (2017). Adapting and controlling DNN-based speech synthesis using input codes. In *Proc. International Conference on Speech and Signal Processing*, pages 4905–4909, New Orleans, USA. IEEE. [Cited in sections 2.1.3 and 4.3.]

Malisz, Z., Berthelsen, H., Beskow, J., and Gustafson, J. (2017). Controlling prominence realisation in parametric DNN-based speech synthesis. In *Proc. Interspeech*, pages 1079–1083, Stockholm, Sweden. [Cited in sections 2.1.3 and 2.2.3.]

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE trans. on pattern analysis and machine intelligence*, 11(7):674–693. [Cited in section 2.2.7.]

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. [Cited in section 6.3.2.2.]

Mendelson, J. and Aylett, M. P. (2017). Beyond the listening test: An interactive approach to tts evaluation. In *Proc. Interspeech*, pages 249–253, Stockholm, Sweden. [Cited in sections 1.1.3, 2.3, and 2.3.3.]

Merritt, T., Putrycz, B., Nadolski, A., Ye, T., Korzekwa, D., Dolecki, W., Drugman, T., Klimkov, V., Moinet, A., Breen, A., et al. (2018). Comprehensive evaluation of statistical speech waveform synthesis. In *Proc. Spoken Language Technology Workshop*, pages 325–331. IEEE. [Cited in section 2.1.]

Mertens, P. (2004). The ProsoGram: Semi-automatic transcription of prosody based on a tonal perception model. In *Proc. Speech Prosody*, Nara, Japan. [Cited in section 2.2.2.]

Mishra, T., Kim, Y.-j., and Bangalore, S. (2015). Intonational phrase break prediction for text-to-speech synthesis using dependency relations. In *Proc. International Conference on Speech and Signal Processing*, pages 4919–4923, Brisbane, Australia. IEEE. [Cited in section 2.2.7.]

Mitchell, R. L. and Ross, E. D. (2013). Attitudinal prosody: What we know and directions for future study. *Neuroscience and Biobehavioral Reviews*, 37(3):471–479. [Cited in sections 1 and 2.2.6.1.]

Mitchell, T. M. (1980). The need for biases in learning generalizations. [Cited in section 2.5.1.]

Mohan, D. S. R., Hu, V., Teh, T. H., Torresquintero, A., Wallis, C. G. R., Staib, M., Foglianti, L., Gao, J., and King, S. (2021). Ctrl-P: Temporal control of prosodic variation for speech synthesis. In *Proc. Interspeech*, Brno, Czech Republic. [Cited in sections 2.1.3, 2.2.7, and 5.2.]

Monrad-Krohn, G. (1947). The prosodic quality of speech and its disorders: (a brief survey from a neurologist's point of view). *Acta Psychiatrica Scandinavica*, 22(3-4):255–269. [Cited in section 2.2.]

Montaño, R. and Alías, F. (2017). The role of prosody and voice quality in indirect storytelling speech: A cross-narrator perspective in four European languages. *Speech Communication*, 88:1–16. [Cited in section 2.2.6.2.]

Mori, M., MacDorman, K. F., and Kageki, N. (2012). The uncanny valley. *IEEE Robotics and Automation Magazine*, 19(2):98–100. [Cited in section 1.]

Morise, M., Yokomori, F., and Ozawa, K. (2016). WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Trans. Inf. Syst.*, 99(7):1877–1884. [Cited in sections 2.1, 2.1.1.1, 2.1.2, 2.2.3, 3.5.1, and 5.4.1.]

Mower, E., Mataric, M. J., and Narayanan, S. (2011). A framework for automatic human emotion classification using emotion profiles. *IEEE Trans. on Audio, Speech, and Language Processing*, 19(5):1057–1070. [Cited in section 4.2.1.]

Nespor, M. and Vogel, I. (2007). *Prosodic phonology*, volume 28. Walter de Gruyter. [Cited in sections 2.2 and 2.2.4.]

Neto, J., Almeida, L., Hochberg, M., Martins, C., Nunes, L., Renals, S., and Robinson, T. (1995). Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system. In *Proc. Eurospeech*. [Cited in section 2.1.3.]

Ng, A. Y. (2004). Feature selection, $L_1$ vs. $L_2$ regularization, and rotational invariance. In *Proc. International Conference on Machine Learning*, page 78. [Cited in section 2.5.1.5.]

Obin, N. (2011). *MeLos: Analysis and modelling of speech prosody and speaking style*. PhD thesis, Université Pierre et Marie Curie-Paris VI. [Cited in section 2.2.2.]

Obin, N., Beliao, J., Veaux, C., and Lacheret, A. (2014). SLAM: Automatic stylization and labelling of speech melody. In *Proc. Speech Prosody*, pages 246–250, Dublin, Ireland. [Cited in sections 1.1.1 and 2.2.2.]

O'Mahony, J., Oplustil-Gallegos, P., Lai, C., and King, S. (2021). Factors affecting the evaluation of synthetic speech in context. In *Proc. Speech Synthesis Workshop*, pages 148–153. [Cited in sections 3 and 7.1.]

Oplustil-Gallegos, P., O'Mahony, J., and King, S. (2021). Comparing acoustic and textual representations of previous linguistic context for improving text-to-speech. In *Proc. Speech Synthesis Workshop*, pages 205–210. [Cited in section 7.1.]

Ostendorf, M. and Veilleux, N. M. (1994). A hierarchical stochastic model for automatic prediction of prosodic boundary location. *Computational Linguistics*, 20(1):27–54. [Cited in section 2.2.7.]

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *J. of Machine Learning Research*, 22(57):1–64. [Cited in section 2.5.2.]

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *Proc. International Conference on Neural Information Processing Systems*, Long Beach, USA. [Cited in sections 1.2.1 and 3.5.1.]

Patton, B., Agiomyrgiannakis, Y., Terry, M., Wilson, K., Saurous, R. A., and Sculley, D. (2016). AutoMOS: Learning a non-intrusive assessor of naturalness-

of-speech. In *Proc. International Conference on Neural Information Processing Systems*, Barcelona, Spain. [Cited in section 2.3.1.]

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572. [Cited in section 5.4.1.1.]

Pell, M. D., Monetta, L., Paulmann, S., and Kotz, S. A. (2009). Recognizing emotions in a foreign language. *Journal of Nonverbal Behavior*, 33(2):107–120. [Cited in section 4.2.1.]

Picard, R. W. and Picard, R. (1997). *Affective computing*, volume 252. MIT press Cambridge. [Cited in sections 1 and 4.2.1.]

Pierrehumbert, J. and Hirschberg, J. B. (1990). The meaning of intonational contours in the interpretation of discourse. [Cited in section 2.2.2.]

Pierrehumbert, J. B. (1980). *The phonology and phonetics of English intonation*. PhD thesis, Massachusetts Institute of Technology. [Cited in sections 2.2.2 and 2.2.4.]

Plutchik, R. (1984). Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984:197–219. [Cited in section 4.2.1.]

Podsiadło, M. and Ungureanu, V. (2018). Experiments with training corpora for statistical text-to-speech systems. In *Proc. Interspeech*, pages 2002–2006, Hyderabad, India. [Cited in sections 2.1.2, 2.4.1, 5, 2.5.1, and 3.3.]

Poria, S., Chaturvedi, I., Cambria, E., and Hussain, A. (2016). Convolutional MKL based multimodal emotion recognition and sentiment analysis. In *Proc. Data Mining (ICDM)*, pages 439–448. IEEE. [Cited in section 4.1.]

Prateek, N., Łajszczak, M., Barra-Chicote, R., Drugman, T., Lorenzo-Trueba, J., Merritt, T., Ronanki, S., and Wood, T. (2019). In other news: A bi-style text-to-speech model for synthesizing newscaster voice with limited data. *arXiv preprint arXiv:1904.02790*. [Cited in sections 2.1.3, 2.4, 2.4.1, 5, 2.5.1, and 4.1.]

Prenger, R., Valle, R., and Catanzaro, B. (2019). WaveGlow: A flow-based generative network for speech synthesis. In *Proc. International Conference on Speech and Signal Processing*, pages 3617–3621. [Cited in section 2.1.2.]

Prevost, S. and Steedman, M. (1994). Specifying intonation from context

for speech synthesis. *Speech Communication*, 15(1-2):139–153. [Cited in section 2.2.6.1.]

Qian, Y., Wu, Z., Gao, B., and Soong, F. K. (2010). Improved prosody generation by maximizing joint probability of state and longer units. *IEEE Trans. on Audio, Speech, and Language Processing*, 19(6):1702–1710. [Cited in section 2.2.7.]

Recommendation P.85, I.-T. (1994). Telephone transmission quality subjective opinion tests. a method for subjective performance assessment of the quality of speech voice output devices. [Cited in section 2.3.2.]

Recommendation P.862, I.-T. (2001). Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. [Cited in section 2.3.1.]

Ren, Y., Hu, C., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. (2020). FastSpeech 2: Fast and high-quality end-to-end text-to-speech. *arXiv preprint arXiv:2006.04558*. [Cited in sections 1.1.2, 1.1.3, 2.1.1.1, 2.1.1.2, 2.1.3, 2.2.7, 2.3, 2.6, 6.1, 6.3.1, 6.4.2, 6.5.3.1, and 6.5.4.]

Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. (2019). FastSpeech: Fast, robust and controllable text to speech. In *Proc. International Conference on Neural Information Processing Systems*, pages 3171–3180, Vancouver, Canada. [Cited in section 2.6.]

Rendel, A., Fernandez, R., Kons, Z., Rosenberg, A., Hoory, R., and Ramabhadran, B. (2017). Weakly-supervised phrase assignment from text in a speech-synthesis system using noisy labels. In *Proc. Interspeech*, pages 759–763, Stockholm, Sweden. [Cited in sections 1.1.1, 2.1.3, and 2.2.7.]

Ribeiro, M. S. (2018a). Parallel audiobook corpus. [dataset]. University of Edinburgh. School of Informatics. doi.org/10.7488/ds/2468. [Cited in section 2.4.2.]

Ribeiro, M. S. (2018b). *Suprasegmental relations for the modelling of fundamental frequency in statistical parametric speech synthesis*. PhD thesis, University of Edinburgh. [Cited in section 2.2.4.]

Ribeiro, M. S. and Clark, R. A. J. (2015). A multi-level representation of F0 using the continuous wavelet transform and the discrete cosine transform. In *Proc.*

*International Conference on Speech and Signal Processing*, pages 4909–4913, Brisbane, Australia. IEEE. [Cited in sections 1.1.1, 2.2.7, 5.2, and 6.2.]

Ribeiro, M. S., Watts, O., and Yamagishi, J. (2017). Learning word vector representations based on acoustic counts. In *Proc. Interspeech*, Stockholm, Sweden. [Cited in section 2.2.6.1.]

Ringeval, F., Schuller, B., Valstar, M., Jaiswal, S., Marchi, E., Lalanne, D., Cowie, R., and Pantic, M. (2015). AV+EC 2015: The first affect recognition challenge bridging across audio, video, and physiological data. In *Proc. Workshop on Audio/Visual Emotion Challenge*, pages 3–8. ACM. [Cited in section 4.2.2.]

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *arXiv preprint arXiv:2002.12327*. [Cited in sections 6.3.2.3, 6.5.3.2, and 7.1.]

Rolfe, J. T. (2016). Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*. [Cited in section 2.2.5.]

Ronanki, S. (2019). *Prosody generation for text-to-speech synthesis*. PhD thesis. [Cited in section 2.2.7.]

Ronanki, S., Henter, G. E., Wu, Z., and King, S. (2016a). A template-based approach for speech synthesis intonation generation using LSTMs. In *Proc. Interspeech*, pages 2463–2467, San Francisco, USA. [Cited in sections 1.1.1, 2.2.7, and 5.2.]

Ronanki, S., Watts, O., King, S., and Henter, G. E. (2016b). Median-based generation of synthetic speech durations using a non-parametric approach. pages 686–692, San Diego, USA. IEEE. [Cited in section 2.2.7.]

Rosenberg, A. (2010). AuToBI - a tool for automatic ToBI annotation. In *Proc. Interspeech*, pages 146–149, Makuhari, Japan. [Cited in sections 2.1.3 and 2.2.7.]

Rosenberg, A., Fernandez, R., and Ramabhadran, B. (2018). Measuring the effect of linguistic resources on prosody modeling for speech synthesis. In *Proc. International Conference on Speech and Signal Processing*, pages 5114–5118, Calgary, Canada. IEEE. [Cited in section 2.2.6.1.]

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information

storage and organization in the brain. *Psychological review*, 65(6):386. [Cited in section 2.5.1.]

Rousselet, G. A., Pernet, C. R., and Wilcox, R. R. (2021). The percentile bootstrap: A primer with step-by-step instructions in R. *Advances in Methods and Practices in Psychological Science*, 4(1). [Cited in section 3.]

Roy, J., Cole, J., and Mahrt, T. (2017). Individual differences and patterns of convergence in prosody perception. *Laboratory Phonology*, 8(1). [Cited in sections 2.1.3, 2.2.2, 2.3.2, and 2.3.3.]

Rozgic, V., Ananthakrishnan, S., Saleem, S., Kumar, R., and Prasad, R. (2012). Ensemble of SVM trees for multimodal emotion recognition. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–4. IEEE. [Cited in section 4.1.]

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical Report Tech. rep. ICS 8504, Institute for Cognitive Science, University of California, San Diego, USA. [Cited in section 2.5.1.2.]

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536. [Cited in sections 2.5.1 and 2.5.1.1.]

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. In *Proc. International Conference on Machine Learning*, pages 1842–1850. PMLR. [Cited in section 7.1.]

Satt, A., Rozenberg, S., and Hoory, R. (2017). Efficient emotion recognition from speech using deep learning on spectrograms. *Proc. Interspeech*, pages 1089–1093. [Cited in section 4.1.]

Scherer, K. R., Banse, R., Wallbott, H. G., and Goldbeck, T. (1991). Vocal cues in emotion encoding and decoding. *Motivation and emotion*, 15(2):123–148. [Cited in section 4.4.4.1.]

Schneider, S., Baevski, A., Collobert, R., and Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*. [Cited in sections 2.2.5 and 6.2.]

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. (2021). Toward causal representation learning. *Proc. IEEE*, 109(5):612–634. [Cited in sections 2.2.5 and 7.1.]

Schröder, M. (2009). Expressive speech synthesis: Past, present, and possible futures. *Affective information processing*, pages 111–126. [Cited in section 2.1.3.]

Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Müller, C., and Narayanan, S. S. (2010). The INTERSPEECH 2010 paralinguistic challenge. [Cited in section 4.1.]

Schuller, B. W., Steidl, S., Batliner, A., Hirschberg, J., Burgoon, J. K., Baird, A., Elkins, A. C., Zhang, Y., Coutinho, E., and Evanini, K. (2016). The INTERSPEECH 2016 computational paralinguistics challenge: Deception, sincerity and native language. In *INTERSPEECH*, pages 2001–2005. [Cited in sections 4.2.2 and 4.1.]

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Trans. on Signal Processing*, 45(11):2673–2681. [Cited in sections 2.5.1.2 and 6.3.1.]

Scordilis, M. S. and Gowdy, J. N. (1989). Neural network based generation of fundamental frequency contours. In *Proc. International Conference on Speech and Signal Processing*, pages 219–222, Glasgow, UK. IEEE. [Cited in section 2.2.7.]

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press. [Cited in section 2.2.1.]

Selkirk, E. O. (1980). *On prosodic structure and its relation to syntactic structure*, volume 194. Indiana University Linguistics Club. [Cited in section 2.2.4.]

Settles, B. (2009). Active learning literature survey. [Cited in section 1.1.2.]

Shattuck-Hufnagel, S. and Turk, A. E. (1996). A prosody tutorial for investigators of auditory sentence processing. *Journal of psycholinguistic research*, 25(2):193–247. [Cited in section 2.2.]

Shechtman, S. and Sorin, A. (2019). Sequence to sequence neural speech synthesis with prosody modification capabilities. In *Proc. Interspeech*, pages 275–280, Graz, Austria. [Cited in section 2.2.7.]

Shen, J., Jia, Y., Chrzanowski, M., Zhang, Y., Elias, I., Zen, H., and Wu, Y.

(2020). Non-attentive Tacotron: Robust and controllable neural TTS synthesis including unsupervised duration modeling. *arXiv preprint arXiv:2010.04301*. [Cited in sections 2.1.1.2, 2.2.7, 2.3.2, 6.1, and 6.6.]

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., Saurous, R. A., Agiomyrgiannakis, Y., and Wu, Y. (2018). Natural TTS synthesis by conditioning wavenet on Mel spectrogram predictions. In *Proc. International Conference on Speech and Signal Processing*, pages 4779–4783, Calgary, Canada. IEEE. [Cited in sections 2.1, 2.1, 2.1.1.2, 2.1.2, 2.2.7, 2.3.2, 2.6, 6.3.1, 6.4, and 6.4.1.]

Sherif, M., Taub, D., and Hovland, C. I. (1958). Assimilation and contrast effects of anchoring stimuli on judgments. *Journal of experimental psychology*, 55(2):150. [Cited in section 2.3.3.]

Sifre, L. (2014). *Rigid-motion scattering for image classification*. PhD thesis, Ecole Polytechnique. [Cited in section 2.5.1.3.]

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., and Hirschberg, J. (1992). ToBI: A standard for labeling English prosody. In *Proc. Spoken Language Processing*. [Cited in sections 1.1.1, 2.2, 2.2.2, 2.2.4, 3.1, and 4.1.]

Sinha, T. and Cassell, J. (2015). We click, we align, we learn: Impact of influence and convergence processes on student learning and rapport building. In *Workshop on modelling INTERPERsonal SynchrONy And influence*, pages 13–20. [Cited in section 2.2.6.2.]

Skerry-Ryan, R., Battenberg, E., Xiao, Y., Wang, Y., Stanton, D., Shor, J., Weiss, R. J., Clark, R., and Saurous, R. A. (2018). Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron. *arXiv preprint arXiv:1803.09047*. [Cited in sections 2.1.1.2, 2.2.5, and 2.2.5.]

Sobel, I. and Feldman, G. (1968). An isotropic 3x3 image gradient operator. [Cited in section 2.5.1.3.]

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. International Conference on Machine Learning*, pages 2256–2265. PMLR. [Cited in section 2.5.2.]

Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. volume 28, pages 3483–3491, Montréal, Canada. [Cited in sections 2.5.2.1 and 3.4.]

Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A., and Bengio, Y. (2017). Char2Wav: End-to-end speech synthesis. [Cited in section 2.1.]

Sproat, R. (2008). Linguistic processing for speech synthesis. *Springer Handbook of Speech Processing*, pages 457–470. [Cited in section 2.1.]

Sproat, R., Black, A. W., Chen, S., Kumar, S., Ostendorf, M., and Richards, C. (2001). Normalization of non-standard words. *Computer Speech and Language*, 15(3):287–333. [Cited in section 2.1.]

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. of Machine Learning Research*, 15(1):1929–1958. [Cited in section 2.5.1.5.]

Stalnaker, R. (1974). Pragmatic presuppositions. page 197–214. New York University Press. [Cited in section 2.2.6.1.]

Stan, A., Watts, O., Mamiya, Y., Giurgiu, M., Clark, R., Yamagishi, J., and King, S. (2013). TUNDRA: A multilingual corpus of found data for TTS research created with light supervision. In *Proc. Interspeech*, pages 2331–2335. [Cited in section 2.4.2.]

Stanton, D., Wang, Y., and Skerry-Ryan, R. (2018). Predicting expressive speaking style from text in end-to-end speech synthesis. *arXiv preprint arXiv:1808.01410*. [Cited in sections 5.2, 6.2, and 6.3.2.]

Steedman, M. (2000). Information structure and the syntax-phonology interface. *Linguistic inquiry*, 31(4):649–689. [Cited in section 2.2.]

Steedman, M. (2014). The surface-compositional semantics of english intonation. *Language*, 90(1):2–57. [Cited in section 2.2.]

Steeneken, H. J. and Houtgast, T. (1980). A physical method for measuring speech-transmission quality. *J. of the Acoustical Society of America*, 67(1):318–326. [Cited in section 2.3.1.]

Stevens, K. N. (2000). *Acoustic phonetics*, volume 30. MIT press. [Cited in sections 2.1.2 and 2.2.3.]

Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *J. of the Acoustical Society of America*, 8(3):185–190. [Cited in section 2.1.2.]

Sun, G., Zhang, Y., Weiss, R. J., Cao, Y., Zen, H., and Wu, Y. (2020). Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis. In *Proc. International Conference on Speech and Signal Processing*, pages 6264–6268. IEEE. [Cited in section 5.2.]

Suni, A., Aalto, D., and Vainio, M. (2015). Hierarchical representation of prosody for statistical speech synthesis. *arXiv preprint arXiv:1510.01949*. [Cited in sections 1.1.1 and 6.2.]

Suni, A., Šimko, J., Aalto, D., and Vainio, M. (2017). Hierarchical representation and estimation of prosody using continuous wavelet transform. *Computer Speech and Language*, 45:123–136. [Cited in section 2.2.7.]

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*. [Cited in section 2.1.1.2.]

Swietojanski, P. and Renals, S. (2014). Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In *Proc. Spoken Language Technology Workshop*, pages 171–176, Lake Tahoe, USA. IEEE. [Cited in section 2.1.3.]

Syrdal, A. K. and McGory, J. (2000). Inter-transcriber reliability of ToBI prosodic labeling. In *Proc. Spoken Language Processing*. [Cited in sections 2.2.2 and 2.3.3.]

Székely, É., Henter, G. E., and Gustafson, J. (2019). Casting to corpus: Segmenting and selecting spontaneous dialogue for TTS with a CNN-LSTM speaker-dependent breath detector. In *Proc. International Conference on Speech and Signal Processing*, pages 6925–6929. IEEE. [Cited in sections 2.2.3, 2.4.2, and 4.1.]

Tachibana, H., Uenoyama, K., and Aihara, S. (2018). Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *Proc. International Conference on Speech and Signal Processing*, pages 4784–4788. IEEE. [Cited in sections 2.1.1.2 and 2.5.1.]

Taghia, J. and Martin, R. (2013). Objective intelligibility measures based on mutual information for speech subjected to speech enhancement processing.

*IEEE Trans. on Audio, Speech, and Language Processing*, 22(1):6–16. [Cited in section 2.3.1.]

Talkin, D. (2015). REAPER: Robust epoch and pitch EstimatoR. github.com/google/REAPER. [Cited in sections 2.2.3 and 5.4.1.]

Taylor, L. and Nitschke, G. (2018). Improving deep learning with generic data augmentation. In *Symposium Series on Computational Intelligence*, pages 1542–1547. IEEE. [Cited in section 2.5.1.5.]

Taylor, P. (1998). The Tilt intonation model. In *Proc. Spoken Language Processing*. [Cited in section 2.2.2.]

Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge university press. [Cited in sections 1, 2.1, and 2.2.3.]

Teixeira, J. P., Oliveira, C., and Lopes, C. (2013). Vocal acoustic analysis – jitter, shimmer and HNR parameters. *Procedia Technology*, 9:1112–1122. [Cited in section 2.2.3.]

Tench, P. (2003). Processes of semogenesis in english intonation. *Functions of Language*, 10(2):209–234. [Cited in section 2.2.6.1.]

Terasawa, H., Slaney, M., and Berger, J. (2005). A timbre space for speech. In *Proc. Interspeech*, pages 1729–1732, Lisbon, Portugal. [Cited in section 2.2.3.]

Tokuda, K., Masuko, T., Yamada, T., Kobayashi, T., and Imai, S. (1995). An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features. In *Proc. Eurospeech*. [Cited in section 2.1.]

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., and Kitamura, T. (2000). Speech parameter generation algorithms for HMM-based speech synthesis. In *Proc. International Conference on Speech and Signal Processing*, volume 3, pages 1315–1318, Istanbul, Turkey. IEEE. [Cited in sections 3.5.1, 3.5.6.3, 4.3.2, and 5.4.1.]

Toledano, D. T., Gómez, L. A. H., and Grande, L. V. (2003). Automatic phonetic segmentation. *IEEE Trans. on Speech and Audio Processing*, 11(6):617–625. [Cited in sections 1.2.1, 2.1.1.1, and 2.2.3.]

Toledano, D. T., Ramos, D., Gonzalez-Dominguez, J., and González-Rodríguez, J. (2009). *Speech Analysis*, pages 1284–1289. Springer. [Cited in section 2.2.3.]

Tomczak, J. M. and Welling, M. (2018). VAE with a VampPrior. In *Proc. Artificial Intelligence and Statistics*, pages 1214–1223, Lanzarote, Spain. [Cited in sections 5.3.2, 5.3.2, and 5.4.1.1.]

Traber, C. (1990). F0 generation with a data base of natural F0 patterns and with a neural network. In *Proc. Speech Synthesis Workshop*, pages 141–144, Autrans, France. [Cited in section 2.2.7.]

Tran, T. (2020). *Neural Models for Integrating Prosody in Spoken Language Understanding*. PhD thesis, University of Washington. [Cited in section 2.2.]

Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M. A., Schuller, B., and Zafeiriou, S. (2016). Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5200–5204. IEEE. [Cited in section 4.2.2.]

Truax, B. (1999). *Handbook For Acoustic Ecology*. Cambridge Street Publishing. [Cited in section 2.2.3.]

Truckenbrodt, H. (2011). The interface of semantics with phonology and morphology. *Semantics: An international handbook of natural language meaning*, 33. [Cited in section 2.2.2.]

Tsuruoka, Y., Miyao, Y., and Jun'ichi, K. (2011). Learning with lookahead: Can history-based models rival globally optimized models? In *Proc. Computational Natural Language Learning*, pages 238–246. [Cited in section 6.3.2.2.]

Turk, A., Nakai, S., and Sugahara, M. (2006). Acoustic segment durations in prosodic research: A practical guide. *Methods in empirical prosody research*, 3:1–28. [Cited in sections 1.1.1 and 2.2.3.]

Turnbull, R., Royer, A. J., Ito, K., and Speer, S. R. (2017). Prominence perception is dependent on phonology, semantics, and awareness of discourse. *Language, Cognition and Neuroscience*, 32(8):1017–1033. [Cited in section 2.3.3.]

Tyagi, S., Nicolis, M., Rohnke, J., Drugman, T., and Lorenzo-Trueba, J. (2020). Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection. In *Proc. Interspeech*, pages 4407–4411, Shanghai, China. [Cited in sections 1.1.3, 2.2.6.1, 2.2.6.2, 2.2.7, 5.2, 6.1, and 6.2.]

Vainio, M., Suni, A., and Aalto, D. (2013). Continuous wavelet transform for analysis of speech prosody. *Proc. TRASP*. [Cited in section 2.2.7.]

Valin, J.-M. and Skoglund, J. (2019). LPCNet: Improving neural speech synthesis through linear prediction. In *Proc. International Conference on Speech and Signal Processing*, pages 5891–5895. [Cited in section 2.1.2.]

Valstar, M., Gratch, J., Schuller, B., Ringeval, F., Lalanne, D., Torres Torres, M., Scherer, S., Stratou, G., Cowie, R., and Pantic, M. (2016). AVEC 2016: Depression, mood, and emotion recognition workshop and challenge. In *Proc. Workshop on Audio/Visual Emotion Challenge*, pages 3–10. ACM. [Cited in section 4.2.2.]

Valstar, M., Schuller, B., Smith, K., Almaev, T., Eyben, F., Krajewski, J., Cowie, R., and Pantic, M. (2014). AVEC 2014: 3d dimensional affect and depression recognition challenge. In *Proc. Workshop on Audio/Visual Emotion Challenge*, pages 3–10. ACM. [Cited in section 4.2.2.]

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*. [Cited in sections 1.2.1, 2.1, 2.1, 2.1.2, 2.5.1.3, and 2.6.]

van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., van den Driessche, G., Lockhart, E., Cobo, L. C., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D., and Hassabis, D. (2018). Parallel WaveNet: Fast high-fidelity speech synthesis. In *Proc. International Conference on Machine Learning*, pages 3918–3926, Stockholm, Sweden. [Cited in sections 2.1.2 and 6.5.2.]

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. (2017). Neural discrete representation learning. In *Proc. International Conference on Neural Information Processing Systems*, pages 6306–6315, Long Beach, USA. [Cited in sections 1.1.1, 2.2.5, 2.2.6.1, 6.2, 6.3.1, and 6.3.2.1.]

van Heuven, V. J., van Bezooijen, R., and Klein WB, P. K. (1995). Quality evaluation of synthesized speech. *Speech coding and synthesis*, pages 707–738. [Cited in section 1.1.3.]

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proc. International Conference on Neural Information Processing Systems*, pages 5998–6008, Long Beach, USA. [Cited in sections 2.5.1.4, 3.5.1, 5.4.1, and 7.1.]

Vilalta, R. and Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95. [Cited in section 2.5.1.4.]

Vinciarelli, A., Pantic, M., and Bourlard, H. (2009). Social signal processing: Survey of an emerging domain. *Image and vision computing*, 27(12):1743–1759. [Cited in section 4.2.1.]

Wagner, M. (2012). Contrastive topics decomposed. *Semantics and Pragmatics*, 5:8–1. [Cited in section 2.2.2.]

Wagner, M. and Watson, D. G. (2010). Experimental and theoretical advances in prosody: A review. *Language and cognitive processes*, 25(7-9):905–945. [Cited in sections 2.2.3 and 2.2.6.]

Wagner, P., Beskow, J., Betz, S., Edlund, J., Gustafson, J., Eje Henter, G., Le Maguer, S., Malisz, Z., Éva Székely, Tånnander, C., and Voße, J. (2019). Speech synthesis evaluation — state-of-the-art assessment and suggestion for a novel research program. In *Proc. Speech Synthesis Workshop*, pages 105–110. [Cited in sections 1, 1.1.3, 2.3, and 2.3.3.]

Wallbridge, S., Bell, P., and Lai, C. (2021). It's not what you said, it's how you said it: discriminative perception of speech as a multichannel communication system. *arXiv preprint arXiv:2105.00260*. [Cited in sections 2.2 and 7.1.]

Wan, V., Chan, C., Kenter, T., Vit, J., and Clark, R. (2019). CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network. In *Proc. International Conference on Machine Learning*, Long Beach, USA. [Cited in sections 1, 1.1.1, 1.1.3, 2.1.3, 2.2.4, 2.2.5, 2.2.5, 2.2.6.1, 2.2.7, 2.5.1.2, 2.6, 3.2, 5.2, and 6.3.1.]

Wang, X. (2018). *Fundamental frequency modelling for neural-network-based statistical parametric speech synthesis*. PhD thesis, SOKENDAI – The Graduate University for Advanced Studies. [Cited in sections 1.1.1, 2.1, 2.2.7, and 5.2.]

Wang, X., Takaki, S., and Yamagishi, J. (2017a). An RNN-based quantized F0

model with multi-tier feedback links for text-to-speech synthesis. pages 1059–1063. [Cited in section 2.2.7.]

Wang, X., Takaki, S., and Yamagishi, J. (2019a). Neural source-filter-based waveform model for statistical parametric speech synthesis. In *Proc. International Conference on Speech and Signal Processing*, pages 5916–5920, Brighton, UK. IEEE. [Cited in section 2.1.2.]

Wang, X., Takaki, S., Yamagishi, J., King, S., and Tokuda, K. (2019b). A vector quantized variational autoencoder (VQ-VAE) autoregressive neural f0 model for statistical parametric speech synthesis. *IEEE Trans. on Audio, Speech, and Language Processing*. [Cited in sections 1.1.1, 2.2.4, 2.2.5, 2.2.5, 2.2.6.1, 2.2.7, 3.2, 5.2, 5.3.1, 6.1, 6.2, 6.3.1, and 6.3.2.]

Wang, Y., Lee, H.-Y., and Chen, Y.-N. (2019c). Tree transformer: Integrating tree structures into self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1061–1070, Hong Kong, China. [Cited in section 2.1.1.2.]

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. (2017b). Tacotron: Towards end-to-end text-to-speech synthesis. *arXiv preprint arXiv:1703.10135*. [Cited in sections 1, 2.1, 2.5.1, and 2.6.]

Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R., Battenberg, E., Shor, J., Xiao, Y., Ren, F., Jia, Y., and Saurous, R. A. (2018a). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *arXiv preprint arXiv:1803.09017*. [Cited in sections 1.1.1, 1.1.3, 2.1.3, 3.2, 5.2, 6.2, and 6.3.1.]

Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R., Battenberg, E., Shor, J., Xiao, Y., Ren, F., Jia, Y., and Saurous, R. A. (2018b). Audio samples from "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis". *google.github.io/tacotron/publications/global_style_tokens/#style_control*. Accessed on 2021-02-10. [Cited in section 2.]

Ward, G. and Hirschberg, J. (1985). Implicating uncertainty: The pragmatics of fall-rise intonation. *Language*, pages 747–776. [Cited in section 2.2.2.]

Ward, N. (2014). Automatic discovery of simply-composable prosodic element. In *Proc. Speech Prosody*, pages 915–919. [Cited in section 2.2.2.]

Ward, N. G. (2019). *Prosodic Patterns in English Conversation*. Cambridge University Press. [Cited in sections 2.2, 2.2.2, 3.1, 5.3.2, 5.4.2.2, and 6.1.]

Ward, N. G., Carlson, J. C., and Fuentes, O. (2018). Inferring stance in news broadcasts from prosodic-feature configurations. *Computer Speech and Language*, 50:85–104. [Cited in section 5.1.]

Watts, O., Henter, G. E., Fong, J., and Valentini-Botinhao, C. (2019). Where do the improvements come from in sequence-to-sequence neural TTS? In *Proc. Speech Synthesis Workshop*, volume 10, pages 217–222, Vienna, Austria. [Cited in sections 2.1, 2.5, and 2.6.]

Watts, O., Henter, G. E., Merritt, T., Wu, Z., and King, S. (2016). From HMMs to DNNs: where do the improvements come from? In *Proc. International Conference on Speech and Signal Processing*, pages 5505–5509, Shanghai, China. IEEE. [Cited in sections 2.5 and 2.6.]

Watts, O., Wu, Z., and King, S. (2015). Sentence-level control vectors for deep neural network speech synthesis. In *Proc. Interspeech*, pages 2217–2221, Dresden, Germany. [Cited in sections 3.2, 3.5.2.2, 4.4.1.2, and 5.2.]

Weiss, R. J., Skerry-Ryan, R., Battenberg, E., Mariooryad, S., and Kingma, D. P. (2020). Wave-tacotron: Spectrogram-free end-to-end text-to-speech synthesis. *arXiv preprint arXiv:2011.03568*. [Cited in sections 2.1, 2.1, and 2.1.1.2.]

Wester, M., Valentini-Botinhao, C., and Henter, G. E. (2015). Are we using enough listeners? no! — an empirically-supported critique of Interspeech 2014 TTS evaluations. In *Proc. Interspeech*, pages 3476–3480, Dresden, Germany. [Cited in section 2.3.2.]

Wester, M., Wu, Z., and Yamagishi, J. (2016). Analysis of the voice conversion challenge 2016 evaluation results. In *Proc. Interspeech*, pages 1637–1641. [Cited in section 2.3.]

Williams, J., Fong, J., Cooper, E., and Yamagishi, J. (2021). Exploring disentan-

glement with multilingual and monolingual VQ-VAE. In *Proc. Speech Synthesis Workshop*, pages 124–129. [Cited in sections 1.1.2, 1.1.3, and 2.1.3.]

Wood, T. and Merritt, T. (2018). Varying speaking styles with neural text-to-speech. Blog article: amazon.science/blog/varying-speaking-styles-with-neural-text-to-speech. Accessed: 2021-04-24. [Cited in section 4.1.]

Wu, B., He, Q., Zhang, P., Koehler, T., Keutzer, K., and Vajda, P. (2020a). FBWave: Efficient and scalable neural vocoders for streaming text-to-speech on the edge. *arXiv preprint arXiv:2011.12985*. [Cited in section 2.1.]

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016a). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*. [Cited in section 6.3.2.3.]

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020b). A comprehensive survey on graph neural networks. *Trans. on neural networks and learning systems*, 32(1):4–24. [Cited in section 7.1.]

Wu, Z., Watts, O., and King, S. (2016b). Merlin: An open source neural network speech synthesis system. In *Proc. Speech Synthesis Workshop*, pages 124–124, Sunnyvale, USA. [Cited in sections 1.2.1, 2.1.1.1, 3.5.1, 4.3.2, and 4.4.3.]

Wu, Z., Xie, Z., and King, S. (2019). The Blizzard challenge 2019. Vienna, Austria. [Cited in sections 2.1 and 2.1.1.2.]

Xia, R. and Liu, Y. (2015). Leveraging valence and activation information via multi-task learning for categorical emotion recognition. In *Proc. International Conference on Speech and Signal Processing*, pages 5301–5305. IEEE. [Cited in section 4.1.]

Xue, S., Abdel-Hamid, O., Jiang, H., Dai, L., and Liu, Q. (2014). Fast adaptation of deep neural network based on discriminant codes for speech recognition. *IEEE Trans. on Audio, Speech, and Language Processing*, 22(12):1713–1725. [Cited in section 3.2.]

Yamagishi, J., Masuko, T., and Kobayashi, T. (2004). HMM-based expressive speech synthesis - towards TTS with arbitrary speaking styles and emotions. In *Proc. of Special Workshop in Maui (SWIM)*. [Cited in section 2.1.3.]

Yamaguchi, A., Chrysostomou, G., Margatina, K., and Aletras, N. (2021). Frustratingly simple pretraining alternatives to masked language modeling. *arXiv preprint arXiv:2109.01819*. [Cited in section 7.1.]

Yamamoto, R., Song, E., and Kim, J.-M. (2019). Probability density distillation with generative adversarial networks for high-quality parallel waveform generation. In *Proc. Interspeech*, pages 699–703. [Cited in section 2.1.2.]

Yasuda, Y., Wang, X., and Yamagishi, J. (2019). Initial investigation of encoder-decoder end-to-end TTS using marginalization of monotonic hard alignments. In *Proc. Speech Synthesis Workshop*, pages 211–216. [Cited in sections 2.1.1.2 and 6.4.2.]

Yu, C., Lu, H., Hu, N., Yu, M., Weng, C., Xu, K., Liu, P., Tuo, D., Kang, S., Lei, G., et al. (2019). DurIAN: Duration informed attention network for multimodal synthesis. *arXiv preprint arXiv:1909.01700*. [Cited in sections 1.1.3, 2.1, 2.1.1.2, 2.1.3, 2.2.7, 2.6, 6.3.1, 6.4, 6.4.2, and 6.5.3.1.]

Yu, K. and Young, S. (2011). Continuous F0 modeling for HMM based statistical parametric speech synthesis. *IEEE Trans. on Audio, Speech, and Language Processing*, 19(5):1071–1079. [Cited in section 2.2.7.]

Yuan, J., Liberman, M., and Cieri, C. (2006). Towards an integrated understanding of speaking rate in conversation. In *Proc. Interspeech*, pages 541–544, Pittsburgh, USA. [Cited in section 2.2.3.]

Zen, H. (2006). An example of context-dependent label format for HMM-based speech synthesis in English. *wiki.inf.ed.ac.uk/twiki/pub/CSTR/F0parametrisation/hts_lab_format.pdf*. Accessed on 2021-04-21. [Cited in section 2.1.1.1.]

Zen, H., Agiomyrgiannakis, Y., Egberts, N., Henderson, F., and Szczepaniak, P. (2016). Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices. In *Proc. Interspeech*, pages 2273–2277, San Francisco, USA. [Cited in sections 2.1, 3.3, and 3.5.6.3.]

Zen, H., Dang, V., Clark, R., Zhang, Y., Weiss, R. J., Jia, Y., Chen, Z., and Wu, Y. (2019). LibriTTS: A corpus derived from LibriSpeech for text-to-speech. *arXiv preprint arXiv:1904.02882*. [Cited in sections 1.1.3, 2.4.2, 4.1, and 5.4.1.1.]

Zen, H. and Senior, A. (2014). Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In *Proc. International Conference on Speech and Signal Processing*, pages 3844–3848, Florence, Italy. IEEE. [Cited in section 3.3.]

Zen, H., Senior, A., and Schuster, M. (2013). Statistical parametric speech synthesis using deep neural networks. In *Proc. International Conference on Speech and Signal Processing*, pages 7962–7966, Vancouver, Canada. IEEE. [Cited in sections 2.1, 2.1, 2.1.1.1, and 2.2.7.]

Zen, H., Tokuda, K., and Black, A. W. (2009). Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064. [Cited in section 2.1.]

Zen, H., Tokuda, K., Masuko, T., Kobayasih, T., and Kitamura, T. (2007). A hidden semi-markov model-based speech synthesis system. *IEICE Trans. on information and systems*, 90(5):825–834. [Cited in section 2.2.7.]

Zhang, R., Atsushi, A., Kobashikawa, S., and Aono, Y. (2017). Interaction and transition model for speech emotion recognition in dialogue. *Proc. Interspeech*, pages 1094–1097. [Cited in section 4.1.]

Zhou, X., Ling, Z.-H., and King, S. (2020). The Blizzard challenge 2020. Shanghai, China. [Cited in sections 2.1.2 and 2.6.]

Zou, X., Bao, X., and Luo, L. (2010). Integration of intonation in F0 trajectory prediction using MSD-HMMs. In *Proc. Speech Prosody*, Chicago, USA. [Cited in section 2.2.7.]

Zou, Y., Liu, S., Yin, X., Lin, H., Wang, C., Zhang, H., and Ma, Z. (2021). Fine-grained prosody modeling in neural speech synthesis using ToBI representation. In *Proc. Interspeech*, pages 3146–3150. [Cited in section 2.1.3.]

Zwicker, E. (1961). Subdivision of the audible frequency range into critical bands (frequenzgruppen). *J. of the Acoustical Society of America*, 33(2):248–248. [Cited in section 2.1.2.]