Bollepalli, Bajibabu; Juvela, Lauri; Airaksinen, Manu; Valentini-Botinhao, Cassia; Alku, Paavo

Normal-to-Lombard adaptation of speech synthesis using long short-term memory recurrent neural networks

Published in:
Speech Communication

DOI:
10.1016/j.specom.2019.04.008

Published: 01/07/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Available online at www.sciencedirect.com

ScienceDirect

Speech Communication 00 (2019) 1–20

Speech Communi-cation

www.elsevier.com/locate/procedia

Normal-to-Lombard Adaptation of Speech Synthesis Using Long Short-Term Memory Recurrent Neural Networks

Bajibabu Bollepalli[a], Lauri Juvela[a], Manu Airaksinen[a], Cassia Valentini-Botinhao[b], Paavo Alku[a]

[a]*Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland.*
[b]*Center for Speech Technology Research, University of Edinburgh, Edinburgh, UK.*

Abstract

In this article, three adaptation methods are compared based on how well they change the speaking style of a neural network based text-to-speech (TTS) voice. The speaking style conversion adopted here is from normal to Lombard speech. The selected adaptation methods are: auxiliary features (AF), learning hidden unit contribution (LHUC), and fine-tuning (FT). Furthermore, four state-of-the-art TTS vocoders are compared in the same context. The evaluated vocoders are: GlottHMM, GlottDNN, STRAIGHT, and pulse model in log-domain (PML). Objective and subjective evaluations were conducted to study the performance of both the adaptation methods and the vocoders. In the subjective evaluations, speaking style similarity and speech intelligibility were assessed. In addition to acoustic model adaptation, phoneme durations were also adapted from normal to Lombard with the FT adaptation method. In objective evaluations and speaking style similarity tests, we found that the FT method outperformed the other two adaptation methods. In speech intelligibility tests, we found that there were no significant differences between vocoders although the PML vocoder showed slightly better performance compared to the three other vocoders.

Keywords:
Lombard, Auxiliary features, LHUC, Fine-tuning, LSTM, Adaptation, TTS

1. Introduction

In noisy environments, human talkers modify their speaking style to make the spoken message more understandable. This phenomenon is called the Lombard effect [32], and the resulting speech is called Lombard speech or speech-in-noise. Both the acoustic and phonetic properties of speech are affected when the speaking style used in a quiet environment is changed to Lombard speech in noisy environment. The main acoustic modifications caused by the Lombard effect are an increase in vocal intensity and fundamental frequency (f_0), a decrease in spectral tilt, and a change in formant frequencies and phoneme durations [62, 27, 23, 22, 33]. Modifications in phonetic properties include, for example, increased prominence in the production of vowels compared to consonants as well as in the production of vowels and consonants compared to semivowels [27, 22].

Present text-to-speech (TTS) synthesis systems have reached a mature state where intelligibility of synthetic speech corresponds to that of natural speech in a quiet environment. However, in real-life applications (such as public address systems, vehicle navigation devices, and mobile phones), TTS is often used in

conditions with severe background noise, and consequently the intelligibility is drastically reduced [11]. Therefore, there is a great need to improve the intelligibility of synthetic speech. This can be done by incorporating the Lombard effect to the development of TTS voices, in a similar manner as natural talkers modify their speaking style in noisy environments. The current study focuses on adaptation to Lombard speech using neural network based TTS systems.

Improvement of speech intelligibility in noise has been studied in many investigations, both in natural (e.g., [73]) and synthetic (e.g., [26]) speech. Several of the previously developed intelligibility improvement methods were evaluated recently in the Hurricane Challenge [10] by conducting a large-scale open evaluation. A total of 14 algorithms were proposed for natural speech, and four systems were proposed for synthetic speech. The performance of the algorithms were evaluated in the Hurricane Challenge using the word error rate (WER) and the equivalent intensity change (EIC), that is, the gain (in dB) by which the level of unmodified speech needs to be raised in order to obtain the same intelligibility score as that of modified speech. The experiments reported in [10, 11] yielded an EIC value of 5.1 dB for natural speech and of 5.6 dB for synthetic speech (using the intelligibility of unmodified synthetic speech as the reference), and reaching up to 37 percentage points of absolute word accuracy improvement.

Most of the existing techniques used for normal-to-Lombard modification of synthetic speech are motivated by the acoustic properties of speech that are affected in the Lombard effect. These techniques apply signal processing methods to mimic the acoustic changes observed in the production of Lombard speech. The methods utilized are cepstral modification using the glimpse proportion measure [59], spectral shaping [18], and dynamic range compression [60]. These techniques typically do not require Lombard speech to modify synthetic speech. However, there are a few studies that explicitly employed Lombard speech to enhance the intelligibility of synthetic speech by using either voice conversion [31] or adaptation techniques [41, 48, 39].

The previous adaptation studies in TTS, however, are all based on statistical parametric speech synthesis (SPSS) systems utilizing hidden Markov model (HMM)-based speech synthesis, due to its adaptation abilities and flexibility in changing voice characteristics (e.g., speaker, speaking style, and emotional category) as well as its small memory footprint [57]. The HMMs trained on normal speech can be adapted with a small amount of Lombard speech data using the technique called constrained structural maximum a posteriori linear regression combined with maximum a posteriori (CSMAPLR + MAP) adaptation [68]. Previous studies have shown that the intelligibility of synthetic speech generated by the Lombard-adapted TTS system is significantly higher in noisy environments than the corresponding synthetic speech of normal speaking style [11, 41, 42].

The quality of HMM-based synthesis is, however, limited by two main factors: 1) accuracy of acoustic modeling, and 2) quality of vocoders. Recently, deep neural networks (DNNs) were proposed as an alternative to HMMs in TTS to improve the accuracy of acoustic modelling. Many independent studies have demonstrated that the quality of synthetic speech generated by DNNs is significantly better than that of HMM-based systems (e.g., [72]). DNNs increase the robustness of acoustic modeling by better capturing the complex dependencies between linguistic and acoustic features. DNNs have been further extended to recurrent neural networks (RNNs), especially long short-term memory networks (LSTMs), to model the sequential nature of speech [21, 71]. To improve the quality of vocoders, neural vocoders (e.g., WaveNet [52]) have been proposed. TTS systems utilizing neural vocoders have yielded high fidelity of synthetic speech, but these techniques also call for large amounts of training data and lots of computational resources. Particularly the first requirement constitutes a severe limitation in synthesis of speaking styles of high vocal effort, such as Lombard speech.

Until now, only a few studies have explored the DNN-based speaker adaptation for TTS, although DNNs have shown promising results in speaker adaptation in the area of speech recognition. In principle, DNNs can be adapted at three levels: input level, model level and output level. At the input level, speaker-specific features, for example i-vectors, are used to augment the conventional textual information [46, 66]. At the model level, the adaptation can be done by scaling the hidden-activation values or by fine-tuning the whole or part of the network with adaptation data [50, 66, 20]. At the output level, a voice transformation technique is typically applied. Using these adaptation methods, a DNN-based system has been shown to outperform an HMM-based adapted system [66]. Recently, adaptation at the input level was utilized

in [65] to adapt a speech synthesizer trained in a declarative speaking style to synthesize speech with an interrogative style. However, to the best of our knowledge, the only previous TTS study on DNN-based Lombard speech adaptation was the one published recently by the present authors [8].

In the current study, normal-to-Lombard adaptation of speech synthesis is studied using deep neural networks. This investigation is a sequel to our pilot study on the topic [8], and it extends the previous one in many respects. Two main research goals are set as follows. The first goal is to find whether a particular vocoder is more suited for the task of normal-to-Lombard adaptation. We hypothesize that some vocoders might provide an acoustic space that is better suited for the task. For instance, if a vocoder provides a parameter for an acoustic property (such as spectral tilt) that is modified by natural talkers when speaking style is changed from normal to Lombard, one would expect the corresponding vocoder to work well also in normal-to-Lombard adaptation of synthetic speech. To reach this goal, we evaluate the following four vocoders: 1) GlottHMM [43], 2) GlottDNN [2], 3) STRAIGHT [29], and 4) PML [14]. There are also other vocoders, such as WORLD [36], AHOCODER [17], Vocaine [1], and MagPhase [19]. However, the chosen vocoders represent the main vocoder types used in current SPSS systems (see [3] for further details): GlottHMM and GlottDNN belong to glottal vocoders, STRAIGHT uses mixed excitation with a spectral envelope, while PML combines an advanced aperiodicity model rooted on sinusoidal vocoding with a log-domain source-filter synthesis model. The second goal is to find what is the most successful adaptation technique for the Lombard speaking style transformation. We hypothesize that some techniques might take benefit of preexisting normal speaking style data and learn Lombard speaking style characteristics from a small amount of data better than the other existing techniques. To reach this goal, we compared the following three adaptation methods: 1) auxiliary features, 2) learning hidden unit contribution, and 3) fine-tuning. Further, the current study conducts an extensive subjective evaluation, for the first time in DNN-based synthesis, in speaking style adaptation by assessing the speaking style similarity between Lombard-adapted synthetic speech and natural Lombard speech as well as between Lombard-adapted synthetic speech and natural speech of normal speaking style. In addition, the study investigates whether there are differences between the vocoding methods selected when they are used in normal-to-Lombard adaptation to improve the intelligibility of of synthetic speech in various noise conditions.

A side goal of this article is to addresses the adaptation of duration model from normal to Lombard speech. In traditional DNN-based SPSS systems, phoneme durations affect synthetic speech in two respects: by changing speech prosody and by changing the feature positions in acoustic modeling. Traditionally in DNN-based SPSS systems, durations are estimated separately from HMMs as neural networks require frame-level mapping, whereas HMMs can segment the utterance at the phoneme level in a weakly supervised manner. However, recent more advanced neural networks, such as sequence-to-sequence models, are able to learn the duration implicitly within models (for example, see Tacotron [64], DeepVoice [6], and Char2Wav [47]). Previous studies on speaker adaptation in TTS have exclusively focused on the acoustic model only. Since variations in durations also contribute to speaker characteristics, it is crucial to also adapt these durations similarly to the acoustic model adaptation. This is particularly important to the two speaking styles addressed in the current study, normal and Lombard, that show large differences in durations (e.g., vowel durations are elongated and consonant durations shortened when speaking style changes from normal to Lombard). Some studies have even indicated that the Lombard effect can be mimicked by only changing the durations [12]. Thus, it is considered crucial to adapt durations in a similar manner as is done in acoustic model adaptation.

2. Statistical parametric speech synthesis (SPSS) system

The goal of a TTS system is to convert a given text input into natural sounding speech. A typical TTS system consists of two main parts, the front-end and the back-end. In the front-end, the text input is converted into a sequence of symbols, called the linguistic specification [53]. The back-end takes advantage of an acoustic model which renders the speech waveform from the linguistic specification generated by the front-end. Although the quality of TTS depends both on the front-end and back-end, this study focuses on the latter. Two paradigms, unit-selection synthesis and statistical parametric speech synthesis (SPSS), are prevalent in the back-end. Figure 1 shows a general block diagram of an SPSS system.

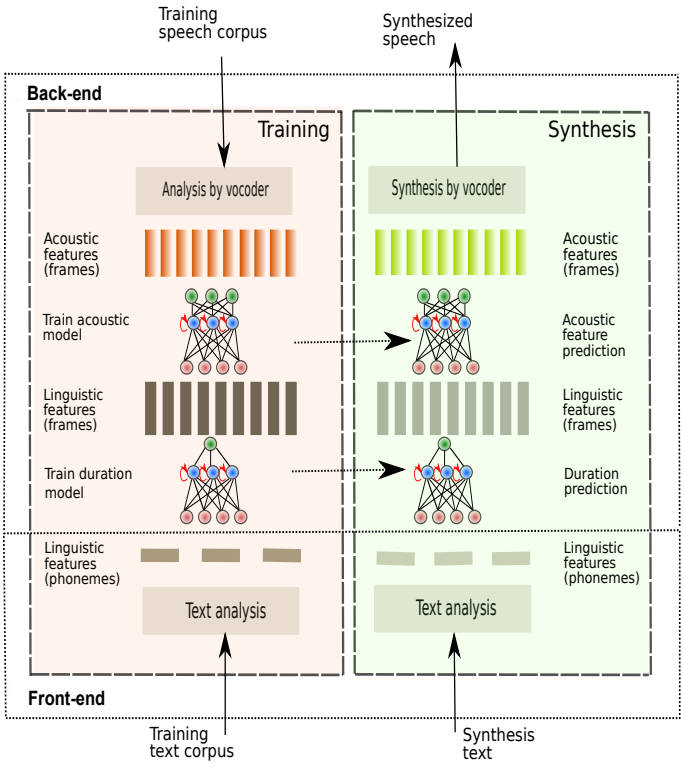

Fig. 1. A schematic diagram of a recurrent neural network based statistical parametric speech synthesis system.

In TTS, the input vectors typically contain many levels of information, such as phoneme, syllable, and words [53]. These linguistic features are language-dependent and extracted by the front-end (for example, the Festival front-end [54] is used for English). The output vectors contain the corresponding acoustic features. The acoustic features are extracted using a vocoder, Section 3 briefly describes a few state-of-the-art vocoders employed in SPSS. In addition, duration information is needed to build a model between text features and acoustic features, since these sequences are of different lengths (e.g., an utterance with 10 phonemes can have 200 frames at 5-ms frame rate).

Conventional DNN-based synthesis systems use a frame or state as the basic modeling unit, which makes it difficult to capture the co-articulation effect. To overcome this shortcoming of DNN-based systems, the use of standard sigmoid layers has been proposed to be replaced with recurrent neural networks (RNN) in TTS [21]. However, conventional RNNs suffer from the vanishing gradient problem, which deteriorates their ability to model long-time relations in sequential features. As a remedy, long short-term memory neural networks (LSTMs) have been proposed [24].

The majority of current neural network based SPSS systems use HMM-based force alignment to model the phoneme duration. The estimated phoneme or state durations are used to interpolate the textual features at the frame level. However, at inference time, it is not always feasible or convenient to first predict durations from HMMs and later use them to predict the acoustic features using LSTM-based systems. To address this problem, a separate duration model was proposed recently in [71], using neural networks in sequence with the acoustic model; thus both acoustic and duration models are in one pipeline.

3. Vocoders

A vocoder is used to express a speech waveform with a parametric representation that can be converted back into a speech waveform. Furthermore, the parametric representation enables the statistical modeling of

speech, and it also makes possible to manipulate speech to enhance its intelligibility. These properties make vocoders flexible tools that can be applied in several areas of speech technology such as statistical parametric speech synthesis, voice transformation and modification, musical applications, and even low bit-rate speech coding. The current study compares vocoders from three main categories (glottal, mixed-excitation, and log-domain source-filter models rooted on sinusoidal vocoding). Altogether four individual vocoders were included, which will be shortly described next.

3.1. GlottHMM

Glottal vocoders aim to parameterize speech according to the functioning of the real human speech production mechanism. The oldest member of glottal vocoders, GlottHMM [43], is based on the source-filter model of speech production that models the speech production mechanism as a cascade of linear time-invariant filters excited by the glottal volume velocity waveform as follows:

$$S(z) = G(z)V(z)L(z), \tag{1}$$

where $S(z)$ is the Z-transform of the speech signal, $G(z)$ is the Z-transform of the glottal excitation, $V(z)$ is the vocal tract transfer function, and $L(z)$ is the transfer function of the lip radiation effect. GlottHMM computes this speech separation by utilizing the iterative adaptive inverse filtering algorithm [5], and then represents $V(z)$ and the spectral tilt of $G(z)$ as autoregressive filters parameterized as line spectral frequencies (LSFs). Furthermore, the fundamental frequency f_0, harmonic-to-noise ratio (HNR), and signal energy are parameterized.

During synthesis, GlottHMM constructs the glottal excitation by modifying and concatenating a manually selected glottal pulse, called the base pulse, according to the $G(z)$ envelope, f_0, HNR, and energy. The unvoiced excitation is generated using white Gaussian noise. Finally, the excitation is filtered with the autoregressive time-variant $V(z)$ filter to obtain the synthesized speech signal.

3.2. GlottDNN

GlottDNN [3] is a more recent glottal vocoder that has been developed as a cumulative evolution from the GlottHMM vocoder. GlottDNN utilizes the same source-filter model as GlottHMM, but has many updated components. Most importantly, the source-filter separation in GlottDNN is computed with the quasi-closed phase analysis glottal inverse filtering method [4]. In the analysis part, the rest of vocoding is done as in GlottHMM.

In the synthesis part, GlottDNN uses a specific DNN to generate glottal excitations. This DNN, which is trained as described in [2], maps the frame-level vocoder feature vector into a corresponding glottal closure instant-centered, two-pitch period long glottal flow derivative waveform. The voiced excitation is generated with the pitch-synchronous overlap-add (PSOLA) [37] procedure. Finally, the generated excitation is filtered with the vocal tract filter $V(z)$ to obtain the synthetic speech signal.

3.3. STRAIGHT

STRAIGHT [29] utilizes the conventional source-filter model of speech where the entire spectral envelope ($S_U(z)$) is driven by a spectrally flat excitation signal ($I(z)$) as follows:

$$S(z) = I(z)S_U(z). \tag{2}$$

STRAIGHT aims to obtain this separation by minimizing the periodicity interference within and between analysis frames (i.e., obtain a smooth spectrogram in both time and frequency axes). This is computed with a pitch-adaptive analysis scheme that utilizes two complementary analysis windows. As the spectral information in STRAIGHT is encoded in the representation of $S_U(z)$, the parameters of speech left for the excitation are f_0 and HNR (called *aperiodicity* in STRAIGHT). STRAIGHT computes the aperiodicity measure for each bin of the magnitude spectrum based on a smoothed table look-up operation that has the ratio of the lower and upper envelope as input and refers to a database of known aperiodicity measurements.

The inherent STRAIGHT parameters are thus f_0 and magnitude spectrograms representing the spectral envelope and aperiodicity. In SPSS, the spectrogram parameters (i.e., frequency bins for each frame) are commonly transformed into mel-generalized cepstral coefficients [56] for the purpose of data compression. The synthetic speech signal is generated in STRAIGHT by synthesizing the minimum phase spectrogram based on the vocoder parameters, and then performing the inverse Fourier transform and overlap-adding the obtained waveforms.

3.4. Pulse model in log-domain (PML)

The PML vocoder [15] utilizes frequency-domain pulse synthesis techniques from parameters that are obtained from sinusoidal analysis. In the analysis part, first a continuous f_0 contour and an estimate of the spectral envelope ($S_U(z)$) is obtained. $S_U(z)$ can be based on the interpolation of harmonic peak amplitudes, or its computation can be out-sourced, for example, to STRAIGHT analysis. Finally, a specific phase distortion deviation (PDD) is computed with the help of harmonic phase distortion (PD) values:

$$\mathrm{PD}_{i,h} = \phi_{i,h+1} - \phi_{i,h} - \phi_{i,1}, \tag{3}$$

where $\phi_{i,h}$ is the phase value at frame i and harmonic h. The PD values are then linearly interpolated to obtain $\mathrm{PD}_i(\omega)$, a continuous spectral representation of phase distortion. $\mathrm{PDD}_i(\omega)$ is then computed as the short-term standard deviation of PD:

$$\mathrm{PDD}_i(\omega) = \sqrt{-2\log\left|\frac{1}{N}\sum_n e^{j(\mathrm{PD}_n(\omega))}\right|}. \tag{4}$$

PDD values show, in a normalized representation, how much phase distortion there is in each frequency bin compared to the estimated f_0. To simplify this model for SPSS, the PDD values are quantized into binary values known as the *binary noise mask* $M_i(\omega)$.

The synthesis part in the PML vocoder is performed in the frequency domain. Based on the (continuous) f_0 contour, a single minimum phase pulse $S_i(\omega)$ of length $\frac{1}{f_0}$ is generated at each pitch mark t_i. The spectrum is set as the minimum phase response of the spectral envelope $V_i(\omega)$, and the phase spectrum values are replaced with random noise at frequency bins where $M_i(\omega) = 1$:

$$S_i(\omega) = e^{-j\omega t_i} \cdot V_i(\omega) \cdot N_i(\omega)^{M_i(\omega)}. \tag{5}$$

where $N_i(\omega)$ is the Fourier transform of Gaussian noise.

4. Adaptation methods

Building a DNN-based standalone TTS system from scratch requires a considerable amount of training data uttered in a specific speaking style. The data scarcity issue for a challenging speaking style (such as Lombard) can, however, be addressed by taking advantage of adaptation and data of another widely available speaking style (such as normal). DNN-based adaptation of TTS is more complex than HMM-based adaptation, since the number of parameters in DNNs is large and the parameters of the DNN model cannot be interpreted as directly as the parameters of the HMM model. In this section, the DNN-based adaptation methods employed in the current study are described. The study takes advantage of three adaptation techniques: 1) auxiliary features (AF), 2) learning hidden unit contribution (LHUC), and 3) fine-tuning (FT). Figure 2 demonstrates the block diagram of each adaptation method employed in the study.

4.1. Auxiliary features (AF)

In this method, the speaking style specific *auxiliary features* are concatenated to the input linguistic features in the training of an acoustic model. Thus, the AF method does not involve any model adaptation per se. The model utilizes the knowledge provided in the input features to discriminate speaking styles. This approach has an advantage that it can be easily applied to a range of deep acoustic models, such as feed

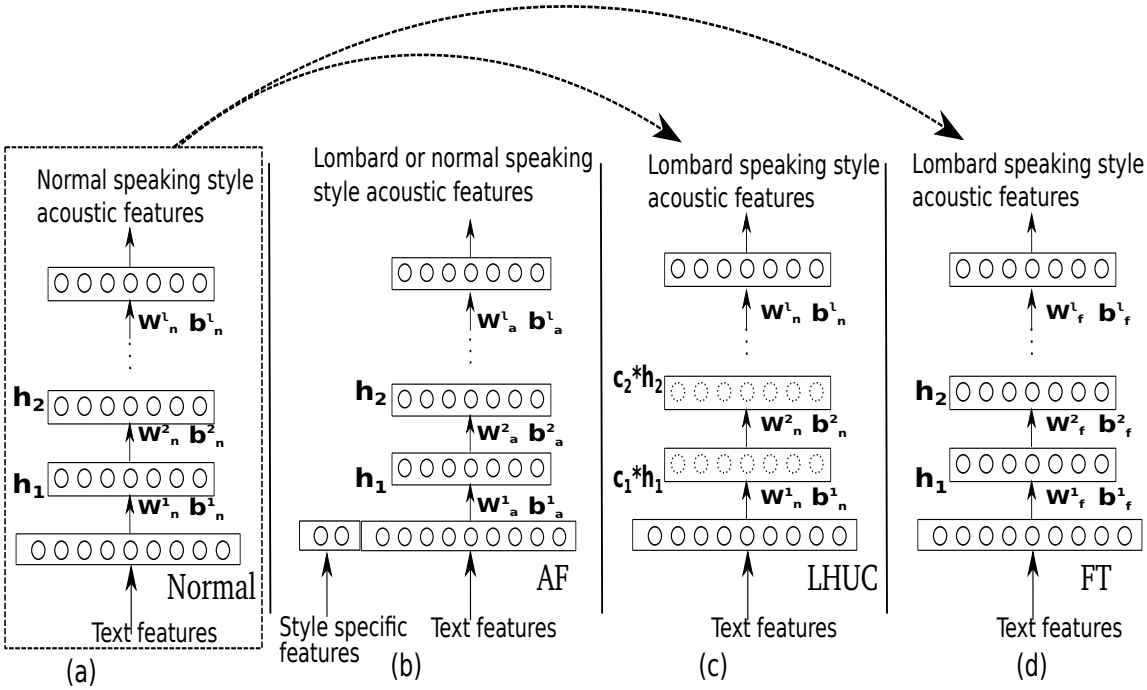

Fig. 2. Illustration of the synthesis systems compared. (a) Normal speaking style TTS system. Adapted systems based on (b) auxiliary features (AF), (c) learning hidden contribution (LHUC), and (d) fine-tuning (FT). Adapted systems LHUC and AF use the normal speaking style TTS system as an initial system.

forward neural networks, LSTM-RNNs, and even waveform generation models (e.g., WaveNet [61]). AF has been used for speaker adaptation in both speech recognition [46] and synthesis [66, 16]. Features like *i-vectors* [66] or *d-vectors* [16] have been used to convey the speaker-specific information in adaptation. Typically, these features are estimated with a separate model, for example, the Gaussian mixture model-universal background model (GMM-UBM) for *i-vector* extraction and another DNN for *d-vector* extraction. Altering these auxiliary features changes the output acoustic features to correspond to the desired target speaking style. One-hot vectors, the simplest representation for speaker variability, have also been studied in this context [35]. Some studies [63] have integrated the learning of auxiliary features into the acoustic model. In the current study, one-hot vectors are used as auxiliary features to capture the style-specific information. Since the study includes two speaking styles, normal and Lombard, two-dimensional one-hot vectors are derived as follows:

$$\begin{aligned} code &= 1 \quad 0 && \text{Normal} \\ &= 0 \quad 1 && \text{Lombard} \end{aligned}$$

Figure 2(b) illustrates adaptation based on using one-hot vectors as auxiliary features, where the style-specific codes are provided along with input features.

4.2. Learning hidden unit contribution (LHUC)

The LHUC method assumes that hidden representations of a neural network are basis vectors learned to represent the acoustic space of many speakers or speaking styles when trained on a large amount of speech data which comprised many speakers and styles. Thus, we can represent new speaking styles by scaling these hidden representations. This method was originally proposed for speaker adaptation in speech recognition with little data [50] and has also been used in speech synthesis for speaker adaptation [66]. However, to the best of our knowledge, this adaptation method has not been previously used in TTS for

speaking style adaptation, except for a pilot study [8] by the current authors. This method has several advantages, including unsupervised speaker adaptation using small data (see, e.g., [49]). First, an SPSS system is built using normal speaking style, and later we borrow parameters from this network to learn a new set of parameters, denoted by $\mathbf{c}$, with Lombard speech, as shown in Figure 2(c). The hidden representations of the SPSS system trained with normal speech are scaled by the newly learned parameters $\mathbf{c}$ as follows:

$$\mathbf{h'}_l = \mathbf{h}_n \circ \mathbf{c} \tag{6}$$

where $\circ$ operation represents the Hadamard product, $\mathbf{h}_n$ represents the hidden representations of normal speech and $\mathbf{h'}_l$ represents the hidden representations of Lombard speech. We can scale all layers or any particular layer based on the task.

4.3. Fine-tuning (FT)

This method falls into the category of transfer learning [38], where the knowledge learned in one task can be used in another similar task. For instance in speech synthesis, one can train an average voice model (AVM) on a large group of speakers and later use the same network to adapt to the new speaker with less data [20]. Since the AVM learns most of the relations between the text and acoustic features, it is easy to shift some of the parameters by training the whole network or a part of it (mostly top layer) with new speaker data. This method, however, might suffer from overfitting. Thus, the cost function is typically regularized with some constraint. The FT method has its origins in the renaissance of deep learning, where deep architectures are pre-trained in an unsupervised manner and later fine-tuned in a supervised fashion. In a few previous TTS studies, the effectiveness of FT has been demonstrated in speaker adaptation [51, 13]. In [51], it was reported that this method outperformed LHUC and an HMM-based adaptation method in terms of both objective and subjective evaluations. The FT adaptation technique used in the current study is slightly different from unsupervised pre-training. Namely, we first train the network on a task with plenty of data (normal speech), and later the same network is trained on a similar task (Lombard speech). This is illustrated in Figure 2, where the normal speaking style TTS system (which is trained first) is shown in Figure 2 (a) and its parameters are $\mathbf{W_n^1}, \mathbf{W_n^2}, \ldots \mathbf{W_n^l}, \mathbf{b_n^1}, \mathbf{b_n^2}, \ldots \mathbf{b_n^l}$. In the adaptation, shown in Figure 2 (d), all the parameters of the normal speaking style system are updated to a new set of parameters $\mathbf{W_f^1}, \mathbf{W_f^2}, \ldots \mathbf{W_f^l}, \mathbf{b_f^1}, \mathbf{b_f^2}, \ldots \mathbf{b_f^l}$ using Lombard speech.

5. Experiments

Multifaceted experiments were designed in the current study in order to compare the four selected vocoders described in Section 3, along with the three adaptation methods described in Section 4 in normal-to-Lombard adaptation of TTS. In this section, the main parts in the design of the experiments are described by first explaining the database and the TTS system used, after which we describe the procedures adopted in the objective and subjective evaluations.

5.1. Database

We employed the Hurricane Challenge database, which can be freely downloaded from the web [9]. The data consists of sentences spoken both in normal and Lombard styles. The speech data was produced by a male professional British voice talent. To elicit the Lombard effect, noise was played in the talker's ears via headphones at a constant level of 84 dB. The text prompts were borrowed from the Harvard phonetic balance text corpus [45]. The number of utterances in the database is 2542 and 720 in normal and Lombard speaking styles, respectively. The data is sampled at 16000 Hz. There was an overlap of 500 utterances in text in the Lombard and normal corpus. For the purposes of the present study, the data was divided into three categories: the training set, the development set, and the test set, as shown in Table 1.

Table 1. Partition of the data (in number of utterances) used in the current study.

Style	Train	Development	Test
Normal	2400	70	72
Lombard	500	100	120

5.2. Systems built

Two types of TTS systems (style-dependent systems and adapted systems) were built using the Merlin speech synthesis toolkit [67] with minor modifications[1]. HMM-based forced alignment was conducted at the state level to get the phoneme durations. The HMMs were trained with the HTS toolkit using the STRAIGHT-based acoustic features. The Festival toolkit was employed to convert the mono-phoneme labels to the full contextual labels. The full contextual labels were mapped onto binary and real values at the frame level using an HTS-style question file. The dimension of linguistic features was 326. A total of nine duration features, containing phoneme state level durations and location of the current frame in the current phoneme, were appended to the linguistic input features. These features were extracted from natural speech at training time from the HMM subphone forced alignment information and predicted from text at synthesis time using the duration model. The resulting total dimension of the input feature vector was 335. The input features were normalized by the min-max normalization technique to the range of [0.01 to 0.99].

Since the study involved four different vocoders, the output features in acoustic modeling depend on the vocoding method used. For STRAIGHT and PML, the spectral envelope was parameterized as mel-generalized cepstral coefficients ($MGCs$) with dimension of 60. For GlottDNN and GlottHMM, the vocal tract filter was parameterized with a linear spectral frequency (LSF) representation with the dimension of 50. Additionally, the glottal vocoders used a filter of the order $m = 10$ to represent the spectral tilt of the glottal excitation (parameterized as $LSFs$). To represent the aperiodicity, the STRAIGHT vocoder used band aperiodicity (BAP), the PML vocoder used phase distortion deviation (PDD), and glottal vocoders used harmonic-to-noise (HNR). The SPTK toolkit [30] was used to extract f_0 values. The same f_0 values were used by all the vocoders in order to avoid discrepancies caused by the f_0 extraction. The f_0 values were linearly interpolated in unvoiced regions, and a binary value was used to keep track of the voiced and unvoiced frames. The acoustic parameters were extracted in 5-ms framerate. The output features consisted of static, delta, and delta-delta features. The mean variance normalization technique was applied to the output features.

LSTM recurrent neural networks were used as acoustic models. The architecture used consisted of three hidden layers followed by a linear layer at the output. The three hidden layers consisted of two feed-forward layers at the bottom and one simplified LSTM layer on top. The bottom feed-forward layers were intended to act as feature extraction layers, with 512 hidden units using the tangent activation function in each layer. The top hidden layer had 256 LSTM blocks. The network parameters were optimized by minimizing the mean square error between the actual and predicted acoustic features using the standard stochastic gradient descent (SGD) optimization algorithm. The initial learning rate was set to a constant value of 0.02 for the first 10 epochs, and afterwards it was decreased by half for each epoch. The mini-batch size was set to 256, and models were trained for 25 epochs with an early stopping criterion.

At synthesis time, the parameters of the test set were predicted from the trained acoustic model. The predicted parameters were further processed by the maximum likelihood parameter generation (MLPG) algorithm [58], and finally, straightforward post-filtering [69] was applied to the spectral features to increase formant dynamics. For GlottDNN and GlottHMM, the spectral valleys of the synthesized vocal tract magnitude envelopes were multiplied by a constant of 0.3 [40]. For PML and STRAIGHT, the values of the first two cepstral coefficients were multiplied by a constant of 1.4 to increase formant dynamics [70].

Five TTS systems (two style-dependent and three adapted) were created for the experiments as follows:

1. **Normal:** The TTS system trained using only speech data of normal speaking style. Two of the

[1] code is uploaded to Merlin `https://github.com/CSTR-Edinburgh/merlin/tree/master/egs/speaker_adaptation`

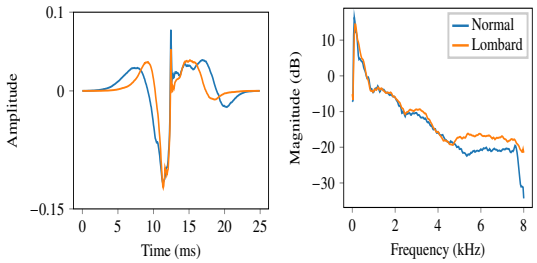

Fig. 3. An example of windowed two-period glottal flow derivative waveforms in both time (left) and frequency (right) domain for normal and Lombard speech.

adaptation methods under comparison (LHUC and FT) use this system as the initial system in the adaptation process.

2. **Lombard:** The standalone Lombard TTS system, where an LSTM-based SPSS system was trained using only Lombard speech data without any adaptation. This system is used as a baseline against which different normal-to-Lombard adaptation systems can be compared.
3. **AF:** The adapted system trained with both Lombard and normal speech, where the speaking style discriminate features are added at the input level.
4. **LHUC:** The adapted system that uses the TTS of normal speaking style as an initial system and Lombard speech to learn the scaling parameters (**c** in Eq. 6).
5. **FT:** The adapted system that uses Lombard speech data to fine-tune all the parameters of the TTS system trained with normal speaking style.

These five systems are developed for each vocoder. Thus, a total of 20 systems (5 systems x 4 vocoders) were built in the experiments.

5.3. Duration model

The durations were modeled using a recurrent LSTM neural network. Initial durations were estimated from HMMs using force alignment. The estimated durations correspond to each monophone and are represented at the state level. The model has two hidden layers. The first layer has 128 linear units with the tangent activation function, and the second layer has 128 LSTM cells. The dimension of input is 326, which contains information on linguistic features as described in 5.2. The dimension of output is 5, which contains the durations of each HMM state. The duration model is trained similarly to the acoustic model described in 5.2.

5.4. Glottal model

Figure 3 illustrates an example of the windowed two-period glottal flow derivative waveform in both time and frequency domain for normal and Lombard speech. Previous studies have shown that the Lombard effect manifests itself also in the spectral tilt of speech [34]. The spectral tilt of speech signals is in turn determined by the characteristics of the glottal excitation. In production of normal speech, the (long-time) spectral envelope of the glottal excitation typically tilts more towards higher frequencies compared to the spectral envelope of the glottal excitation in Lombard speech, as shown in Figure 3. Since two of the vocoders included in the current study (GlottHMM and GlottDNN) use glottal pulses as synthesis excitation waveforms, modelling the glottal model in terms of its spectral tilt was considered justified. The GlottHMM vocoder uses a pre-computed, representative glottal pulse and a few other source-related acoustic features to create the excitation signal. GlottDNN uses instead a deep neural network to compute the time-domain glottal waveform from acoustic parameters. In the current study, we developed a separate glottal model using DNNs, as described in [8].

5.5. Objective evaluations

Due to the large scale of the study, it is not feasible to conduct subjective speech intelligibility tests for all the systems. Therefore, we conducted objective evaluations to reduce the number of systems to be compared. Objective evaluations were carried out on the test set by computing the following metrics[2]:

- **MCD:** Mel-cepstral deviation to measure MGC prediction performance in STRAIGHT and PML.
- **BAP:** A distortion measure for BAPs in STRAIGHT.
- **PDD:** A distortion measure for PDDs in PML.
- **LSF, LSFsource:** A distortion measure for LSFs in GlottHMM and GlottDNN.
- **HNR:** A distortion measure for HNRs in GlottHMM and GlottDNN.
- **Gain:** A distortion measure for Gains in GlottHMM and GlottDNN.
- $f_0 - RMSE$**:** Root mean square error (RMSE) to measure the prediction of f_0. We note that f_0 was modelled on a *log scale*, but the error was calculated on *a linear scale*.
- $f_0 - CORR$**:** Correlation between the predicted and actual f_0 contours.
- **VUV:** Voiced/unvoiced error.

The error metrics chosen are widely used in TTS evaluations (e.g., [8, 66]). In all the metrics, a low score indicates better performance except in $f_0 - CORR$, for which a high score indicates better performance. Natural speech durations were used to create the predicted features so that original and predicted sequences were aligned.

5.6. Speaking style similarity test

In TTS style adaptation, the goal is to convert synthetic speech of one style (source style, i.e., normal speaking style in the current study) to resemble speech spoken in another style (target style, i.e., Lombard in the current study) without sacrificing the quality or speaker similarity of the corresponding synthetic voice. In the literature, various types of similarity tests have been conducted (for example, see the Blizzard test evaluations in [7]). To be used in the evaluation of the Lombard-adapted speech signals of the current study, we modified the setup used in [55] to measure speaker similarity. Each stimulus consists of two utterances, a natural speech signal (either normal or Lombard) and a synthesized signal. The subjects were asked to compare the second utterance to the first one following this instruction:

Do you think that these two samples have been produced using the same speaking style? Some of the samples may sound slightly distorted. Please try to ignore the distortion and concentrate on identifying the speaking style. You have four options to indicate your opinion.

- *Speaking style sounds the same, I'm absolutely sure*
- *Speaking style sounds the same, but I'm not completely sure*
- *Speaking style sounds different, but I'm not completely sure*
- *Speaking styles sounds different, I'm absolutely sure*

[2] https://github.com/CSTR-Edinburgh/merlin/blob/master/src/utils/compute_distortion.py

The speaking style similarity test was conducted on a crowdsourcing platform, CrowdFlower, since it is hard to find an adequate number of native speakers of English in the country where this study was conducted (Finland). The test was made available only to English-speaking countries and countries which have a high rank in English proficiency website. In order to separate representative listeners from outliers, a pre-screening test was conducted, where the subjects must get at least 90% of the answers correct. The samples in the pre-screening test were built to be easily recognizable by using original speech and highly distorted signals. A total of 13 utterances were employed in the pre-screening test. A total of 10 sentences were selected from the test set to evaluate the each adaptation method. Each utterance was rated by 50 listeners. The order of the test sentences was randomized for each subject automatically.

5.7. Speech intelligibility test

Since Lombard speech is used in natural communication situations to enhance speech intelligibility in noisy conditions, it is important to evaluate the different adaptation methods in terms of speech intelligibility. For this purpose, we followed the Hurricane Challenge setup [10]. Speech intelligibility tests are more laborious compared to speaking style similarity tests. Thus, we used for each vocoder only the best adaptation technique among the three adaptation methods (AF, LHUC, and FT) based on their performance in objective evaluations and style similarity tests. Two noise conditions were created to corrupt synthetic speech signals in the intelligibility test, using the same noise signals from [10]. The first one was a stationary, speech-shaped noise (SSN) condition, where the long-term average speech spectrum of a female speaker was used to model the noise spectrum envelope. The second one was a non-stationary, competing speaker (CS) noise condition, where a female speaker interferes with the synthetic male voice. In each noise condition, three SNRs were employed: high SNR (snrHi), medium SNR (snrMid), and low SNR (snrLo). The SNR values in the SSN condition were -1dB, -4dB, and -9dB, whereas in the CS condition, the SNRs were -7dB, -14dB, and -21dB, following the numbers chosen in [10]. To create a stimulus, the speech signal was scaled according to the required SNR value and summed with the corresponding noise signal. The test was conducted in a quiet listening booth by playing the noise-corrupted speech stimuli to the listeners' ears via headphones. In the speech intelligibility test, 24 systems were evaluated: 4 vocoders (each combined with the best adaptation method) x 3 SNRs x 2 noise conditions. For each combination, 5 utterances were used. Thus, each listener evaluated a total of 120 utterances. These 120 utterances were divided into 6 blocks. Each block consisted of 20 utterances with the same noise condition in order to avoid discomfort from noise condition changes during a block. Note that the intelligibility test was designed to compare the four vocoders when they are used with the best adaptation method in various noise conditions, but the test did not compare the Lombard-adapted systems against the normal speaking style TTS system. The normal speaking style TTS system was not included because: (1) there are previous studies indicating that normal-to-Lombard adaptation improves intelligibility in TTS [10, 26] and (2) we wanted to keep the test time for each listener under 1 hour.

The test was conducted in single-walled listening booths with a background noise level less than 10 dB in the frequency range of the test samples. Listeners used circumaural Sennheiser HD650 headphones and they were allowed to adjust the loudness to a comfortable level in a small practice session before the test, after which the volume setting was kept unchanged throughout the intelligibility test. Each stimulus was presented once. Listeners were instructed to type on a computer screen what they had heard irrespective of grammatical structure, after which the subsequent stimulus was presented. The test user interface used in the test is same as in the Hurricane Challenge. The test required around 45 minutes to complete per listener. A total of 13 native English-speaking subjects participated in the test.

5.8. Pilot test of the WaveNet vocoder

A separate, small-scale preliminary experiment was carried out in order to demonstrate the naturalness that is achieved with the WaveNet vocoder when this vocoder is trained with the small amount of Lombard speech available in the current study. In other words, instead of including WaveNet directly to the formal similarity and intelligibility tests described above, we decided to first conduct an informal test to

Table 2. Objective evaluation results of the STRAIGHT and PML vocoders.

	System	MCD (dB)	BAP (dB)	f_0-RMSE (Hz)	f_0-CORR	VUV (%)
STRAIGHT	Normal-Normal	4.301	2.196	9.890	0.801	2.62
	Normal-Lombard	7.339	2.587	45.377	0.773	4.113
	Lombard	5.311	2.124	16.5	0.844	3.79
	AF	5.366	2.147	20.17	0.765	3.177
	LHUC	5.951	2.289	19.504	0.767	4.16
	FT	**4.88**	**2.06**	**15.72**	**0.858**	**2.73**
	System	MCD (dB)	PDD (dB)	f_0-RMSE (Hz)	f_0-CORR	VUV (%)
PML	Normal-Normal	4.298	1.064	9.908	0.801	3.044
	Normal-Lombard	7.359	1.281	45.657	0.772	4.471
	Lombard	5.320	1.054	17.431	0.823	4.024
	AF	5.602	1.075	22.201	0.713	4.591
	LHUC	5.889	1.237	19.524	0.762	4.521
	FT	**4.869**	**1.0**	**16.072**	**0.855**	**2.784**

Table 3. Objective evaluation results of the GlottDNN and GlottHMM vocoders.

	System	Gain (dB)	LSF	LSFsource	HNR	f_0-RMSE (Hz)	f_0-CORR	VUV (%)
GlottDNN	Normal-Normal	2.609	0.182	0.142	6.49	9.838	0.805	2.326
	Normal-Lombard	4.687	0.284	0.187	11.308	45.991	0.763	3.252
	Lombard	3.339	0.209	0.150	8.787	17.002	0.834	3.101
	AF	3.231	0.207	0.147	8.694	18.299	0.812	2.402
	LHUC	3.881	0.226	0.157	9.561	18.491	0.782	3.07
	FT	**3.148**	**0.199**	**0.143**	**8.415**	**15.621**	**0.859**	**2.234**
	System	Gain (dB)	LSF	LSFsource	HNR	f_0-RMSE (Hz)	f_0-CORR	VUV (%)
GlottHMM	Normal-Normal	2.576	0.176	0.052	4.814	9.867	0.804	2.336
	Normal-Lombard	4.671	0.303	0.079	9.338	46.382	0.765	3.270
	Lombard	3.375	0.205	0.064	7.145	17.159	0.831	3.424
	AF	3.626	0.217	0.067	7.562	23.107	0.691	3.345
	LHUC	3.841	0.227	0.072	7.933	18.878	0.773	3.334
	FT	**3.13**	**0.195**	**0.060**	**6.808**	**15.753**	**0.857**	**2.229**

evaluate whether the naturalness achieved with WaveNet is at all comparable to that of the other vocoders studied. We used a WaveNet configuration similar to [28], i.e., three repetitions of a 10-layer convolution stack with exponentially growing dilations, 64 residual channels, and 128 skip channels. The resulting receptive field was 3071 samples. The model was trained using 8-bit categorical cross entropy on quantized $\mu-$law companded signals. The STRAIGHT acoustic parameters (*MGC*, *BAP*, and $lf0$) were used in local conditioning. Since the amount of Lombard data available to train the WaveNet vocoder was little (i.e., 30 mins), the synthetic speech samples generated were, as expected, unintelligible (the samples are available for listening at `http://tts.org.aalto.fi/lombard_wavenet/`). Thus, we decided not to include the samples generated by the WaveNet vocoder in our formal experiments.

6. Results

6.1. Results of objective evaluations

Objective evaluation results are shown for STRAIGHT and PML in Table 2 and for GlottDNN and GlottHMM in Table 3. Rows correspond to six different systems built for this evaluation, which are described as follows. The acoustic parameters predicted by the TTS system trained with normal speaking style only (denoted as Normal in 5.2) were compared with the corresponding parameters extracted from natural normal speech and natural Lombard speech; these results are denoted, respectively, as Normal-Normal and Normal-Lombard in the tables. The row denoted as Lombard includes objective scores computed between the acoustic parameters predicted by the standalone Lombard TTS system (denoted as Lombard in 5.2) and

the corresponding parameters extracted from natural Lombard speech. The remaining rows include objective scores computed between the acoustic parameters predicted by the three Lombard-adapted systems and the corresponding parameters extracted from natural Lombard speech. The name of the row denotes the adaptation method employed.

As expected, the results computed using natural Lombard speech as reference (Normal-Lombard) are much worse than the results obtained by using natural normal speech as reference (Normal-Normal) when the acoustical parameters were predicted with a TTS system trained with normal speaking style. By comparing the three adaptation systems, it can be seen that FT gave the best result for all vocoders and in all metrics. The baseline TTS system Lombard performed slightly better than AF and LHUC in the mel-cepstral distortion (MCD), line spectral frequency distortion (LSF), and line spectral frequency distortion of glottal source ($LSFsource$) metrics. LHUC performed worst in the MCD, LSF, and $LSFsource$ metrics, whereas AF performed worst in the fundamental frequency ($f_0 - RMSE$), f_0 correlation ($f_0 - CORR$), and voiced and unvoiced error (VUV) metrics.

In glottal vocoders, GlottHMM showed lower distortion scores than GlottDNN with the exception of $f_0 - RMSE$ and $f_0 - CORR$, whereas the PML vocoder received lower MCD values compared to STRAIGHT.

The f_0 and VUV features are common for all the vocoders and are extracted using the SPTK toolkit. The differences in the $f_0 - RMSE$ and VUV error metrics are very small across all the vocoders. These differences can result from model initializations. Interestingly, all the vocoders showed a similar trend in the objective evaluations, i.e, FT is the best followed by Lombard, AF, and LHUC. The best scores showed by the Lombard-adapted systems are still far from the corresponding scores achieved by the TTS system of normal speech (Normal-Normal), which indicates that there is still room for improvement.

6.2. Results of subjective evaluations

6.2.1. Speaking style similarity test

Figure 4 shows the results of the style similarity test for each vocoder. Here we show the results when comparing the five TTS system to natural Lombard speech (left) and to natural normal speech (right). The vocoders are separated into four rows, and in each figure the five systems are separated into columns. The motivation to compare the adapted samples to normal speech was to analyze whether the adaptation methods are able to produce the Lombard effect into the synthesized speech. In the ideal case, the Lombard-synthesized speech should sound clearly different from the natural normal speech and, vice-versa, the synthesized samples of normal speaking style should be easily distinguished from natural Lombard speech. This ideal case manifests itself as a general trend in Figure 4: the blue bars of Lombard, AF, LHUC, and FT in the right panes are high, and the blue bars of Normal in the left panes are also high. As in the objective evaluations, FT performed best among the proposed adaptation and baseline systems. Most of the listeners found speech produced by FT sounding similar to natural Lombard speech and different from natural normal speech. This adaptation method performed best in all vocoders. The other two adaptation systems (AF and LHUC) performed more or less equal to the baseline Lombard system.

Statistical significances of the similarity test were analyzed using the non-parametric Mann-Whitney U-test, as recommended in [44]. The results of the statistical tests are shown in the Appendix Tables A.7, A.8, A.9 and A.10. This data shows that the FT method performed significantly better than the two other adaptation methods (AF and LHUC) and also significantly better compared to the baseline (Lombard) system when the reference was natural Lombard speech.

In order to better describe vocoder differences for the best performing adaptation method (i.e., FT), the results of the speaking style similarity test are shown separately for each of the four vocoders in Figure 5. The bars on the left show that a slightly larger number of 'Same' responses were given to the Lombard-adapted samples produced by the PML vocoder than to those generated by the other three vocoders. This difference, however, was not statistically significant, as shown in Table A.11. When comparing the Lombard-adapted synthetic samples w.r.t natural normal speech (Figure 5, right pane), GlottHMM shows the lowest performance, and this degradation is also significantly different from the other vocoders (see Table A.11). This indicates that the listeners did not pay as much attention to the subtle differences between vocoders when Lombard-adapted synthetic speech was compared to the natural Lombard reference. However, when

Table 4. Speech intelligibility scores in WER (%) of the FT adaptation method in competing speaker (CS) noise condition.

Vocoder	snrHi	snrMid	snrLo
GlottDNN	27.2	49.1	79.3
GlottHMM	30.0	**43.8**	79.5
PML	**26.5**	53.0	**77.0**
STRAIGHT	27.7	46.6	79.9

Table 5. Speech intelligibility scores in WER (%) of the FT adaptation method in speech shaped (SSN) noise condition.

Vocoder	snrHi	snrMid	snrLo
GlottDNN	14.5	31.9	74.7
GlottHMM	13.5	28.4	72.6
PML	**13.4**	31.8	72.0
STRAIGHT	13.9	**27.7**	**71.0**

the reference was switched to natural normal speech, one of the vocoders (GlottHMM) was significantly worse.

6.2.2. Speech intelligibility test

The results of the speech intelligibility test are shown in Table 4 and Table 5. Due to the laborious nature of the test, the number of adaptation methods was limited to one per vocoder. We used FT as the adaptation method for each vocoder since it obtained the best scores in both the objective evaluations as well as in the subjective style similarity tests. Table 4 shows the word error rate (WER) in the competing speaker (CS) noise condition, and Table 5 shows WER in the speech-shaped noise (SSN) condition. In computing WER, we removed the most common words such as *a, an, the, in, to, on, is, and, of, for, and at* and manually corrected the spelling mistakes. WERs were computed using a Python package called 'wer' [3].

We can observe that the WERs obtained in the SSN condition are lower than in the more challenging CS noise condition. In the SSN condition, the PML vocoder achieved the best WER in both the snrHi and snrLo conditions, whereas GlottHMM achieved the best WER in the snrMid condition. In the CS noise condition, GlottHMM, PML, and STRAIGHT perform equally well in snrHi, but in snrMid, GlottHMM performed best, and in snrLo STRAIGHT performed best. Overall, the PML vocoder gave the best WER scores in both noise conditions.

To evaluate the intelligibility test significance, we performed a 3-way ANOVA for vocoder, noise type, SNR, and all their interactions. The main effects for noise type and SNR, as well as their interaction were statistically significant with $p < 0.05$ ($F = 69.74, 452.98, 6.029$, respectively). For vocoders, neither the main effect nor any interaction effects were significant.

6.3. Results of duration adaptation

Since the FT method showed the best performance both in all objective tests and in all subjective style similarity evaluations, we used this method for duration adaptation. Table 6 shows the duration adaptation scores. We computed RMSE (frames/phoneme) between the actual durations, which are estimated using force-alignment from HMMs and the predicted durations on the test set. It is not surprising that the scores obtained for the TTS system Normal are very high, as the reference is duration extracted from Lombard speech. The next row shows the improved scores obtained for the standalone TTS system Lombard. The lowest objective scores were given by the FT method.

[3] https://github.com/belambert/asr-evaluation [last accessed in July, 2018]

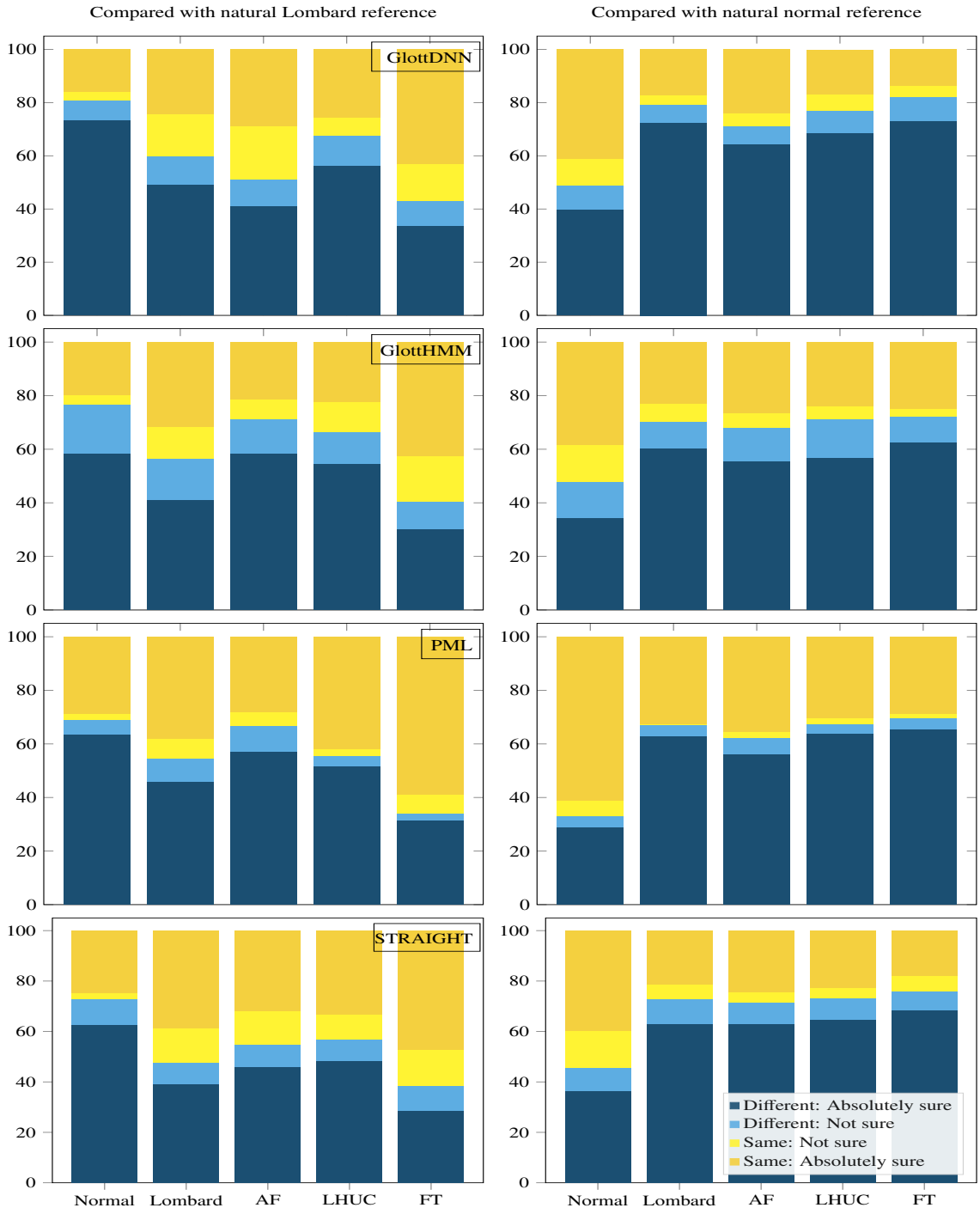

Fig. 4. Results of the style similarity test for the GlottDNN (first row), GlottHMM (second row), PML (third row), and STRAIGHT (fourth row) vocoder. The left column shows the results compared to the natural Lombard reference, and the right column shows the results compared to the natural normal reference.

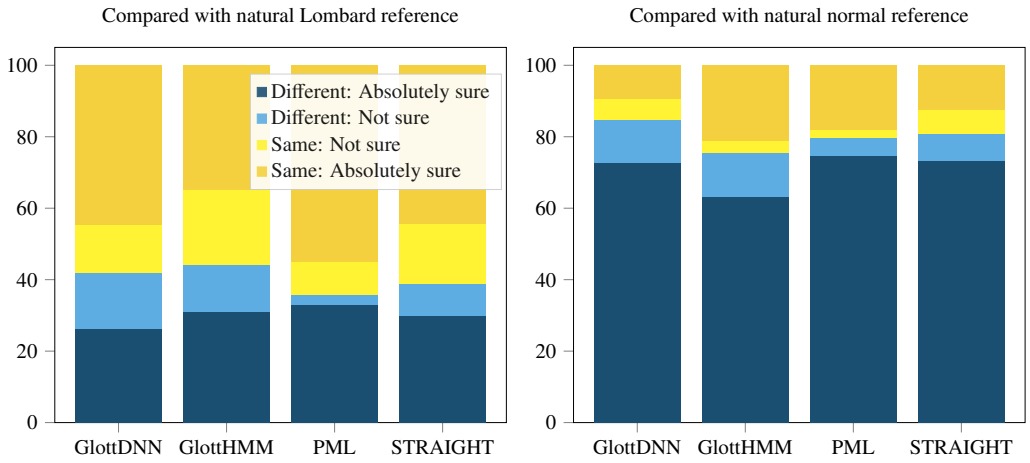

Fig. 5. Results of the style similarity test. For each vocoder, the FT adaptation method was used.

Table 6. Objective scores of duration model adaptation using the FT method.

System	RMSE (Frames)	CORR
Normal	9.691	0.618
Lombard	7.951	0.737
FT	**6.970**	**0.805**

7. Summary and Conclusions

There is a great need for TTS systems that are capable of improving the intelligibility of synthetic speech in noisy environments. An effective technique to improve the intelligibility of synthetic speech in noise is to adapt the synthesizer's speaking style from normal to Lombard, as shown in previous studies on HMM-based TTS systems [41, 48, 39]. This study investigated normal-to-Lombard adaptation of neural network based TTS. The DNN-based adaptation methods studied utilized auxiliary features (AF), learning hidden unit contribution (LHUC), and fine-tuning (FT), which were originally introduced for speaker adaptation in both speech synthesis and recognition. Since vocoding is known to affect the quality of synthetic speech in SPSS, the current study also investigated the role of vocoding in normal-to-Lombard adaptation. The vocoders selected were: two glottal vocoders (GlottHMM and GlottDNN), one mixed excitation vocoder (STRAIGHT), and one log-domain source-filter vocoder rooted on sinusoidal vocoders (PML). To evaluate the performance of each adaptation and vocoder method, we conducted both objective tests and two types of subjective evaluations (a speaking style similarity test and a speech in noise intelligibility test). In the similarity test, apart from the three adapted systems, two baseline systems trained with normal and Lombard speech were also included. All combinations of systems and vocoders were evaluated. In the speech intelligibility test, we compared the different vocoders using the FT adaptation method only. For this evaluation, the synthetic speech was corrupted with speech-shape noise and competing speaker noise at three different SNR levels.

In the objective evaluation, we found that one of the proposed adaptation methods, FT, performed better than the baseline Lombard system. In the speaking style similarity test, we again found that the FT adaptation method performed better than the baseline Lombard system. Both normal and Lombard style data were spoken by the same speaker which means that those two sets of data share many similarities. We believe this helped the FT method to outperform the other two (AF and LHUC) adaptation methods. With respect to the vocoders, the results of the speaking style similarity test revealed that there were statistically significant differences between the vocoders when comparing the adapted synthetic Lombard speech to natural normal speech (the GlottHMM vocoder being significantly worse than the other vocoders). There was, however, no

statistically significant differences between the vocoders when comparing style similarity of the Lombard-adapted synthetic utterances to the corresponding natural Lombard utterances. In the speech intelligibility tests, there was no significant differences between the four vocoders evaluated. For future work, we will experiment with the adaptation of TTS to other speaking styles and in addition use sequence-to-sequence based networks for modeling both acoustic and duration models in a single model. Since the performance of vocoders depend upon the voice [3] and speaking style [25], we aim to investigate whether the observations made in the current study can be generalized to Lombard speech of multiple speakers.

References

[1] Agiomyrgiannakis, Y., 2015. Vocaine the vocoder and applications in speech synthesis, in: ICASSP, IEEE. pp. 4230–4234.

[2] Airaksinen, M., Bollepalli, B., Juvela, L., Wu, Z., King, S., Alku, P., 2016. GlottDNN-a full-band glottal vocoder for statistical parametric speech synthesis, in: Interspeech, pp. 2473–2477.

[3] Airaksinen, M., Juvela, L., Bollepalli, B., Yamagishi, J., Alku, P., 2018. A comparison between STRAIGHT, glottal, and sinusoidal vocoding in statistical parametric speech synthesis. IEEE/ACM Trans. on Audio, Speech, and Language Proc. 26, 1658–1670.

[4] Airaksinen, M., Raitio, T., Story, B., Alku, P., 2014. Quasi closed phase glottal inverse filtering analysis with weighted linear prediction. IEEE/ACM Trans. on Audio, Speech, and Language Proc. 22, 596–607.

[5] Alku, P., 1992. Glottal wave analysis with pitch synchronous iterative adaptive inverse filtering. Speech Communication 11, 109–118.

[6] Arik, S.O., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Ng, A., Raiman, J., et al., 2017. Deep voice: Real-time neural text-to-speech. arXiv preprint arXiv:1702.07825 .

[7] Black, A.W., Tokuda, K., 2005. The Blizzard challenge-2005: Evaluating corpus-based speech synthesis on common datasets, in: Proc. 9th European Conference on Speech Communication and Technology.

[8] Bollepalli, B., Airaksinen, M., Alku, P., 2017. Lombard speech synthesis using long short-term memory recurrent neural networks, in: Proc. ICASSP, IEEE. pp. 5505–5509.

[9] Cooke, M., Mayo, C., Valentini-Botinhao, C., 2013a. Hurricane natural speech corpus. [sound].

[10] Cooke, M., Mayo, C., Valentini-Botinhao, C., 2013b. Intelligibility-enhancing speech modifications: the Hurricane challenge., in: Proc. INTERSPEECH, pp. 3552–3556.

[11] Cooke, M., Mayo, C., Valentini-Botinhao, C., Stylianou, Y., Sauert, B., Tang, Y., 2013c. Evaluating the intelligibility benefit of speech modifications in known noise conditions. Speech Communication 55, 572–585.

[12] Cooke, M., Mayo, C., Villegas, J., 2014. The contribution of durational and spectral changes to the Lombard speech intelligibility benefit. The Journal of the Acoustical Society of America 135, 874–883.

[13] Cooper, E., Hirschberg, J., 2018. Adaptation and frontend features to improve naturalness in found-data synthesis. Proc. Speech Prosody 2018 .

[14] Degottex, G., Lanchantin, P., Gales, M., 2016. A pulse model in log-domain for a uniform synthesizer, in: Proc. 9th ISCA Speech Synthesis Workshop, pp. 230 – 236.

[15] Degottex, G., Lanchantin, P., Gales, M., 2018. A log domain pulse model for parametric speech synthesis. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 26, 57–70.

[16] Doddipatla, R., Braunschweiler, N., Maia, R., 2017. Speaker adaptation in DNN-based speech synthesis using d-vectors. Proc. INTERSPEECH , 3404–3408.

[17] Erro, D., Sainz, I., Navas, E., Hernaez, I., 2014a. Harmonics plus noise model based vocoder for statistical parametric speech synthesis. IEEE Journal of Selected Topics in Signal Processing 8, 184–194.

[18] Erro, D., Zorilă, T.C., Stylianou, Y., 2014b. Enhancing the intelligibility of statistically generated synthetic speech by means of noise-independent modifications. IEEE/ACM Trans. on Audio, Speech, and Language Proc. 22, 2101–2111.

[19] Espic, F., Botinhao, C.V., King, S., 2017. Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis, in: Proc. Interspeech 2017, pp. 1383–1387.

[20] Fan, Y., Qian, Y., Soong, F.K., He, L., 2015. Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis, in: Proc. ICASSP, IEEE. pp. 4475–4479.

[21] Fan, Y., Qian, Y., Xie, F.L., Soong, F.K., 2014. TTS synthesis with bidirectional LSTM based recurrent neural networks., in: Proc. INTERSPEECH, pp. 1964–1968.

[22] Garnier, M., Bailly, L., Dohen, M., Welby, P., Lœvenbruck, H., 2006. An acoustic and articulatory study of Lombard speech: Global effects on the utterance, in: Proc. 9th International Conference on Spoken Language Processing.

[23] Hansen, J.H., 1996. Analysis and compensation of speech under stress and noise for environmental robustness in speech recognition. Speech Communication 20, 151–173.

[24] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9, 1735–1780.

[25] Hu, Q., Richmond, K., Yamagishi, J., Latorre, J., 2013. An experimental comparison of multiple vocoder types, in: Eighth ISCA Workshop on Speech Synthesis.

[26] Huang, D.Y., Rahardja, S., Ong, E.P., 2010. Lombard effect mimicking, in: Proc. 7th ISCA Workshop on Speech Synthesis.

[27] Junqua, J.C., 1993. The Lombard reflex and its role on human listeners and automatic speech recognizers. The Journal of the Acoustical Society of America 93, 510–524.

[28] Juvela, L., Tsiaras, V., Bollepalli, B., Airaksinen, M., Yamagishi, J., Alku, P., 2018. Speaker-independent raw waveform model for glottal excitation, in: Proc. Interspeech 2018, pp. 2012–2016.

[29] Kawahara, H., Estill, J., Fujimura, O., 2001. Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT, in: Proc. MAVEBA.
[30] Kobayashi, T., Tokuda, K., Masuko, T., Koishida, K., et al., 2009. Speech signal processing toolkit (SPTK), version 3.3.
[31] Langner, B., Black, A.W., 2005. Improving the understandability of speech synthesis by modeling speech in noise, in: Proc. ICASSP, IEEE. pp. 265–268.
[32] Lombard, E., 1911. Le signe d'élévation de la voix [the sign of the elevation of the voice]. Annales des maladies de l'oreille et du larynx 37, 101–119.
[33] Lu, Y., Cooke, M., 2008. Speech production modifications produced by competing talkers, babble, and stationary noise. The Journal of the Acoustical Society of America 124, 3261–3275.
[34] Lu, Y., Cooke, M., 2009. The contribution of changes in F0 and spectral tilt to increased intelligibility of speech produced in noise. Speech Communication 51, 1253–1262.
[35] Luong, H.T., Takaki, S., Henter, G.E., Yamagishi, J., 2017. Adapting and controlling DNN-based speech synthesis using input codes, in: Proc. ICASSP, IEEE. pp. 4905–4909.
[36] Morise, M., Yokomori, F., Ozawa, K., 2016. WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE Transactions on Information and Systems 99, 1877–1884.
[37] Moulines, E., Charpentier, F., 1990. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. Speech Communication 9, 453–467.
[38] Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22, 1345–1359.
[39] Picart, B., Drugman, T., Dutoit, T., 2014. Analysis and HMM-based synthesis of hypo and hyperarticulated speech. Computer Speech & Language 28, 687–707.
[40] Raitio, T., Suni, A., Pulakka, H., Vainio, M., Alku, P., 2010. Comparison of formant enhancement methods for HMM-based speech synthesis, in: Proc. 7th ISCA Speech Synthesis Workshop (SSW7).
[41] Raitio, T., Suni, A., Vainio, M., Alku, P., 2011a. Analysis of HMM-based Lombard speech synthesis., in: Proc. INTERSPEECH, pp. 2781–2784.
[42] Raitio, T., Suni, A., Vainio, M., Alku, P., 2014. Synthesis and perception of breathy, normal, and Lombard speech in the presence of noise. Computer Speech & Language 28, 648–664.
[43] Raitio, T., Suni, A., Yamagishi, J., Pulakka, H., Nurminen, J., Vainio, M., Alku, P., 2011b. HMM-based speech synthesis utilizing glottal inverse filtering. IEEE Trans. on Audio, Speech, and Language Proc. 19, 153–165.
[44] Rosenberg, A., Ramabhadran, B., 2017. Bias and statistical significance in evaluating speech synthesis with mean opinion scores, in: Proc. INTERSPEECH, pp. 3976–3980.
[45] Rothauser, E., 1969. Ieee recommended practice for speech quality measurements. IEEE Trans. on Audio and Electroacoustics 17, 225–246.
[46] Saon, G., Soltau, H., Nahamoo, D., Picheny, M., 2013. Speaker adaptation of neural network acoustic models using i-vectors., in: Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 55–59.
[47] Sotelo, J., Mehri, S., Kumar, K., Santos, J.F., Kastner, K., Courville, A., Bengio, Y., 2017. Char2wav: End-to-end speech synthesis. Proc. International Conference on Learning Representations (ICLR) .
[48] Suni, A., Karhila, R., Raitio, T., Kurimo, M., Vainio, M., Alku, P., 2013. Lombard modified text-to-speech synthesis for improved intelligibility: submission for the Hurricane challenge 2013, in: Proc. INTERSPEECH, pp. 3562–3566.
[49] Swietojanski, P., Li, J., Renals, S., 2016. Learning hidden unit contributions for unsupervised acoustic model adaptation. IEEE/ACM Trans. on Audio, Speech, and Language Proc. 24, 1450–1463.
[50] Swietojanski, P., Renals, S., 2014. Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models, in: Proc. Spoken Language Technology Workshop (SLT), IEEE. pp. 171–176.
[51] Takaki, S., Kim, S., Yamagishi, J., 2016. Speaker adaptation of various components in deep neural network based speech synthesis, in: Proc. 9th ISCA Speech Synthesis Workshop (SSW9), pp. 153–159.
[52] Tamamori, A., Hayashi, T., Kobayashi, K., Takeda, K., Toda, T., 2017. Speaker-dependent WaveNet vocoder, in: Proc. Interspeech, pp. 1118–1122.
[53] Taylor, P., 2009. Text-to-speech synthesis. Cambridge university press.
[54] Taylor, P., Black, A.W., Caley, R., 1998. The architecture of the Festival speech synthesis system, in: Proc. 3rd ESCA Workshop on Speech Synthesis.
[55] Toda, T., Chen, L.H., Saito, D., Villavicencio, F., Wester, M., Wu, Z., Yamagishi, J., 2016. The voice conversion challenge 2016., in: Proc. INTERSPEECH, pp. 1632–1636.
[56] Tokuda, K., Kobayashi, T., Masuko, T., Imai, S., 1994. Mel-generalized cepstral analysis - a unified approach to speech spectral estimation, in: Proc. 3rd International Conference on Spoken Language Processing.
[57] Tokuda, K., Nankaku, Y., Toda, T., Zen, H., Yamagishi, J., Oura, K., 2013. Speech synthesis based on hidden Markov models. Proceedings of the IEEE 101, 1234–1252.
[58] Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., Kitamura, T., 2000. Speech parameter generation algorithms for HMM-based speech synthesis, in: Proc. ICASSP, IEEE. pp. 1315–1318.
[59] Valentini-Botinhao, C., Yamagishi, J., King, S., Maia, R., 2014. Intelligibility enhancement of HMM-generated speech in additive noise by modifying mel-cepstral coefficients to increase the glimpse proportion. Computer Speech & Language 28, 665–686.
[60] Valentini-Botinhao, C., Yamagishi, J., King, S., Stylianou, Y., 2013. Combining perceptually-motivated spectral shaping with loudness and duration modification for intelligibility enhancement of HMM-based synthetic speech in noise., in: Proc. INTERSPEECH, pp. 3567–3571.
[61] Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499 .
[62] Van Summers, W., Pisoni, D.B., Bernacki, R.H., Pedlow, R.I., Stokes, M.A., 1988. Effects of noise on speech production:

Acoustic and perceptual analyses. The Journal of the Acoustical Society of America 84, 917–928.
[63] Wan, M., Degottex, G., Gales, M.J., 2017. Integrated speaker-adaptive speech synthesis, in: Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 705–711.
[64] Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R.J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., et al., 2017. Tacotron: Towards end-to-end speech synthesis. arXiv preprint arXiv:1703.10135 .
[65] Wu, X., Sun, L., Kang, S., Liu, S., Wu, Z., Liu, X., Meng, H., 2018. Feature based adaptation for speaking style synthesis, in: Proc. ICASSP, IEEE. pp. 5304–5308.
[66] Wu, Z., Swietojanski, P., Veaux, C., Renals, S., King, S., 2015. A study of speaker adaptation for DNN-based speech synthesis, in: Proc. INTERSPEECH.
[67] Wu, Z., Watts, O., King, S., 2016. Merlin: An open source neural network speech synthesis system, in: Proc. 9th ISCA Speech Synthesis Workshop (SSW9).
[68] Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., Isogai, J., 2009. Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. IEEE Trans. on Audio, Speech, and Language Proc. 17, 66–83.
[69] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T., 2005a. Incorporating a mixed excitation model and postfilter into HMM-based text-to-speech synthesis. Systems and Computers in Japan 36, 43–50.
[70] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T., 2005b. Incorporating a mixed excitation model and postfilter into hmm-based text-to-speech synthesis. Systems and Computers in Japan 36, 43–50.
[71] Zen, H., Sak, H., 2015. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis, in: Proc. ICASSP, IEEE. pp. 4470–4474.
[72] Zen, H., Senior, A., Schuster, M., 2013. Statistical parametric speech synthesis using deep neural networks, in: Proc. ICASSP, IEEE. pp. 7962–7966.
[73] Zorila, T.C., Kandia, V., Stylianou, Y., 2012. Speech-in-noise intelligibility improvement based on spectral shaping and dynamic range compression, in: Proc. INTERSPEECH.

Appendix A. Statistical significance tests of style similarity results

Table A.7. U-test p-values for the different systems using the GlottDNN vocoder

Lombard ref \ Normal ref	FT	Lombard	LHUC	AF	Normal
FT		1.0	0.220	0.976	**0.000**
Lombard	**0.000**		1.0	1.0	**0.000**
LHUC	**0.000**	1.0		1.0	**0.000**
AF	0.924	**0.017**	**0.008**		**0.000**
Normal	**0.000**	**0.000**	**0.000**	**0.000**	

Table A.8. U-test p-values for the different systems using the GlottHMM vocoder

Lombard ref \ Normal ref	FT	Lombard	LHUC	AF	Normal
FT		1.0	1.0	**0.048**	**0.000**
Lombard	**0.000**		1.0	0.626	**0.000**
LHUC	**0.000**	**0.022**		0.082	**0.000**
AF	**0.000**	**0.000**	0.263		**0.000**
Normal	**0.000**	**0.000**	**0.003**	0.914	

Table A.9. U-test p-values for the different systems using the PML vocoder

Lombard ref \ Normal ref	FT	Lombard	LHUC	AF	Normal
FT		1.0	1.0	0.146	**0.000**
Lombard	**0.008**		1.0	0.445	**0.000**
LHUC	**0.000**	0.056		**0.040**	**0.000**
AF	**0.000**	**0.040**	1.0		**0.000**
Normal	**0.000**	**0.000**	**0.001**	**0.005**	

Table A.10. U-test p-values for the different systems using the STRAIGHT vocoder

Lombard ref \ Normal ref	FT	Lombard	LHUC	AF	Normal
FT		1.0	1.0	1.0	**0.000**
Lombard	0.135		1.0	1.0	**0.000**
LHUC	**0.000**	**0.017**		1.0	**0.000**
AF	**0.000**	0.257	1.0		**0.000**
Normal	**0.000**	**0.000**	**0.000**	**0.000**	

Table A.11. U-test p-values with Bonferroni correction for multiple comparisons. Pairwise differences to a Lombard reference were not found statistically significant, whereas GlottHMM stood out as significantly worse when compared to the normal style reference.

Lombard ref \ Normal ref	GlottDNN	GlottHMM	PML	STRAIGHT
GlottDNN		**0.017**	1.0	1.0
GlottHMM	0.421		**0.003**	**0.032**
PML	1.0	0.195		1.0
STRAIGHT	1.0	0.377	1.0	