Artificial Speech Bandwidth Extension Using Deep Neural Networks for Wideband Spectral Envelope Estimation

Johannes Abel and Tim Fingscheidt, *Senior Member, IEEE*

***Abstract*—Estimating a wideband spectral envelope having only narrowband speech at hand is a challenging task. In this paper, we explore ways to do so in the context of an artificial speech bandwidth extension (ABE) framework. Starting from a typical hidden Markov model (HMM)/Gaussian mixture model baseline scheme, we investigate two types of features, topologies, and regularization approaches of deep neural networks (DNNs) to obtain estimates of wideband spectral envelopes with smallest cepstral distance to the original ones. In order to draw realistic conclusions, we employ a database for test, which is acoustically different to the training and validation speech material. Interestingly, it turns out that a DNN regression approach outperforms all other investigated methods, although the HMM has been dropped. Cepstral distance was reduced by 1.18 dB, wideband PESQ was improved by 0.23 MOS points, and a subjective comparison category rating listening test showed a significant preference of the best DNN ABE approach versus narrowband speech of 1.37 CMOS points.**

***Index Terms*—Artificial speech bandwidth extension, deep neural networks, speech enhancement.**

I. Introduction

ARTIFICIAL bandwidth extension (ABE) is a speech enhancement technique being employed on the receiver side of a telephony call. In general, transmitted speech in telephony calls is limited w.r.t. its acoustical bandwidth and thus, speech quality and intelligibility suffers. In case of a narrowband (NB) call, frequency components up to 4 kHz are transmitted (sampling rate of $f'_s = 8$ kHz), while in a wideband (WB) call, frequency components up to 8 kHz are transmitted (sampling rate of $f_s = 16$ kHz). In several studies, the perceived speech quality is shown to substantially increase if an upper band (UB) ($4 < f \leq 8$ kHz) is available [1], [2]. Please note that in a WB call the lower cut-off frequency of down to 50 Hz additionally contributes to the highly improved speech quality. Tackling the lack of acoustical bandwidth in NB calls, ABE approaches typically estimate missing UB spectral components of the transmitted far-end speech signal first, and append then the estimated spectral components to the underlying NB speech signal. As a result, both speech quality and intelligibility of the received speech are enhanced [3]–[5].

The majority of ABE approaches makes use of the source-filter model for speech production. Hereby, the task of extending the NB speech signal is split into finding an UB residual signal and an UB spectral envelope individually. While an UB residual signal is typically estimated from the NB residual signal using rather simple modulation techniques such as spectral folding [6], the estimation of a suitable UB envelope is a rather challenging task. To model the connection between a NB speech signal and an UB spectral envelope, in [7]–[9] two codebooks are jointly trained, one containing NB speech representations and the other one the respective UB envelopes. To classify which UB envelope should be used for synthesis, the minimal distance to the current NB speech frame is identified and the respective UB envelope or UB parameter set is chosen. Furthermore, Gaussian mixture models (GMMs) [10], [11] and hidden Markov models (HMMs) [12]–[15] have been employed as underlying models for classification/estimation with codebooks containing UB spectral envelopes. Supporting HMM-based classification, neural networks show good performance for fricative detection in ABE [16], which is beneficial information for UB estimation [13], [17]. Following the source-filter model, Kontio, Pulakka, and colleagues find the UB residual signal via spectral folding, while a neural network controls a time-variant spline function shaping the UB residual [18], [19].

Hinton *et al.* have shown that the use of feed-forward deep neural networks (DNNs) instead of GMMs as acoustic model for HMMs improve phoneme recognition [20]. Inspired by this performance gain, DNNs have been applied to several fields of speech processing, including ABE. In [21], [22], several statistical models were investigated, such as Gaussian-binary restricted Boltzman machines (RBMs), conditional RBMs, (deep) autoencoders, and sum-product networks for estimation of log spectrograms. Also ignoring the source-filter model, DNNs are used in [23], [24] for directly estimating frequency components. Furthermore, an estimation of the UB log-power spectrum was conducted using a recurrent neural network with long short-term memory (LSTM) cells [25]. In addition, a vocoder-based approach implementing bidirectional associative memories was evaluated in [26].

Manuscript received April 6, 2017; revised August 11, 2017; accepted September 28, 2017. Date of publication October 9, 2017; date of current version November 29, 2017.The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhen-Hua Ling. *(Corresponding author: Tim Fingscheidt.)*

The authors are with the Institute for Communications Technology, Technische Universität Braunschweig, 38106 Braunschweig, Germany (e-mail: j.abel@tu-bs.de; t.fingscheidt@tu-bs.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2017.2761236

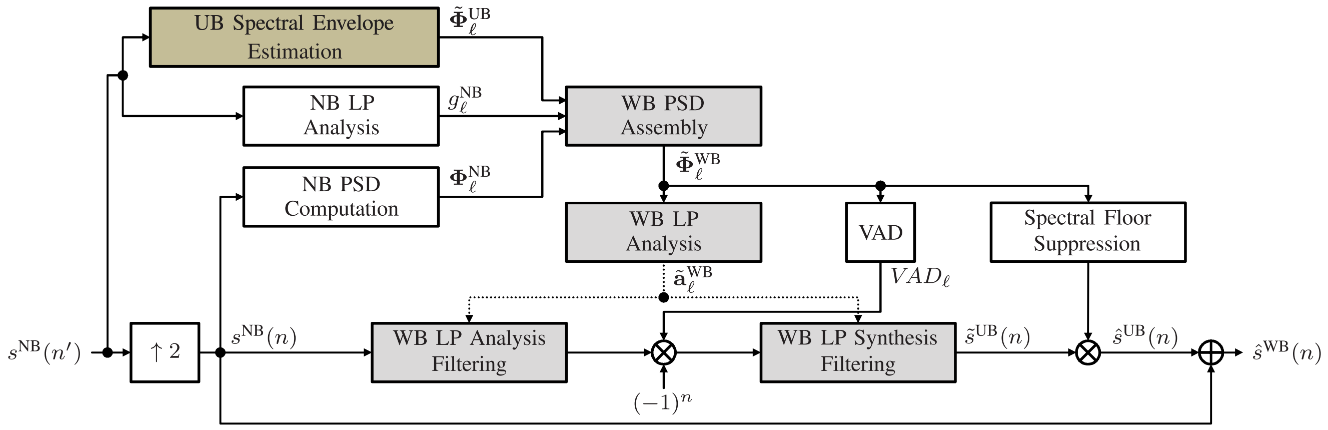
UB Spectral Envelope Estimation
$\tilde{\mathbf{\Phi}}_\ell^{\mathrm{UB}}$
NB LP Analysis
g_ℓ^{NB}
WB PSD Assembly
NB PSD Computation
$\mathbf{\Phi}_\ell^{\mathrm{NB}}$
$\tilde{\mathbf{\Phi}}_\ell^{\mathrm{WB}}$
WB LP Analysis
VAD
Spectral Floor Suppression
$\tilde{\mathbf{a}}_\ell^{\mathrm{WB}}$
VAD_ℓ
$s^{\mathrm{NB}}(n')$
↑ 2
$s^{\mathrm{NB}}(n)$
WB LP Analysis Filtering
WB LP Synthesis Filtering
$\tilde{s}^{\mathrm{UB}}(n)$
$\hat{s}^{\mathrm{UB}}(n)$
$\hat{s}^{\mathrm{WB}}(n)$
$(-1)^n$

Fig. 1. Detailed block diagram of the *artificial bandwidth extension framework*. The input signal $s^{\mathrm{NB}}(n')$ with sampling rate $f_s' = 8$ kHz is artificially extended providing the artificial WB speech signal $\hat{s}^{\mathrm{WB}}(n)$ sampled at $f_s = 16$ kHz as output.

Back to source-filter model-based ABE, DNN-based spectral envelope estimation for ABE is conducted in [27], [28], however, both works adopting quite ad-hoc network topologies and parameter choices.

DNNs have a variety of free parameters and training options, which need to be examined in order to find an optimal statistical model for ABE. Many current ABE-related publications are lacking a thorough investigation of the influence of these parameters on ABE performance.

In this work, we will evaluate the employment of feed-forward DNNs for UB spectral envelope estimation by investigating the influence of a variety of parameters: Topology of the network, employed units, type of inputs, type of targets, dropout as regularization method, and pretraining of network weights. The conducted DNN experiments are compared to a typical HMM/GMM-based estimation scheme. We use several speech databases for training, validation, and test of the UB spectral estimators to obtain a highly practice-oriented experimental setup and conclusions.

The paper is structured as follows: First, an overview to the employed ABE framework is given in Section II. In more detail, Section III will present the employed techniques for UB spectral estimation, including feature extraction, GMM- and DNN-based classification schemes and furthermore also a DNN-based regression approach. Subsequently, Section IV will explain the experimental setup by defining data sets, preprocessing of the speech data, and training of the statistical models. Also metrics for instrumentally assessing the estimated spectral envelopes are defined. In Section V a variety of training options is presented and investigated. The best performing DNN-driven UB spectral estimator is then instrumentally and subjectively assessed w.r.t. speech quality in Section VI. Conclusions are drawn in Section VII.

II. Artificial Bandwidth Extension Framework

Fig. 1 shows a block diagram depicting the ABE framework. The NB speech signal $s^{\mathrm{NB}}(n')$, with n' being the 8 kHz sample index, is input to the framework [13]. The UB speech signal $\hat{s}^{\mathrm{UB}}(n)$ is then estimated frame-by-frame, with n being the 16 kHz sample index, and finally the artificially-extended speech signal $\hat{s}^{\mathrm{WB}}(n) = s^{\mathrm{NB}}(n) + \hat{s}^{\mathrm{UB}}(n)$ is provided, with $s^{\mathrm{NB}}(n)$ being the upsampled and low-pass-filtered input signal.

By means of the source-filter model for speech production, the task of finding the UB speech signal $\hat{s}^{\mathrm{UB}}(n)$ is divided into finding an UB residual signal and an UB spectral envelope individually. The general principle of this ABE framework is to obtain an (interpolated) NB residual signal from the interpolated input signal $s^{\mathrm{NB}}(n)$ via WB linear prediction (LP) analysis filtering and shifting of the spectral content of the NB residual signal to the UB via modulation with $(-1)^n$ [6]. Final WB LP synthesis filtering of the approximated UB residual signal then delivers the estimated UB speech signal $\tilde{s}^{\mathrm{UB}}(n)$. Obviously, the filter coefficients for WB LP analysis and synthesis need to be estimated beforehand.

For frame ℓ, LP analysis and synthesis filtering are performed using the same WB filter coefficients $\tilde{\mathbf{a}}_\ell^{\mathrm{WB}}$. These filter coefficients represent the spectral envelope of the underlying NB speech signal as well as the estimated UB spectral envelope. Therefore, they are perfectly suited for both, LP analysis filtering of the interpolated NB input signal $s^{\mathrm{NB}}(n)$, and LP synthesis filtering of the approximated UB residual signal to obtain an estimated UB speech signal. The WB LP filter coefficients are calculated from the WB power spectrum density (PSD) representation $\tilde{\mathbf{\Phi}}_\ell^{\mathrm{WB}}$, which is assembled from the NB PSD $\mathbf{\Phi}_\ell^{\mathrm{NB}}$ and the estimated UB envelope PSD $\tilde{\mathbf{\Phi}}_\ell^{\mathrm{UB}}$. While the NB PSD is easily obtained using the absolute squared short-term discrete Fourier transform (DFT) from the interpolated NB input signal $s^{\mathrm{NB}}(n)$, estimation of a suitable UB envelope PSD $\tilde{\mathbf{\Phi}}_\ell^{\mathrm{UB}}$ is not trivial, is core of this work, and will be explained in Section III along with the role of g_ℓ^{NB}. Once the WB PSD $\tilde{\mathbf{\Phi}}_\ell^{\mathrm{WB}}$ is composed, auto-correlation coefficients are obtained via inverse DFT and final application of the well-known Levinson-Durbin recursion [29] leads to the desired filter coefficients $\tilde{\mathbf{a}}_\ell^{\mathrm{WB}}$.

A voice activity detection (VAD) as presented in [30], [16] is used in *all* investigated baseline and new approaches, both in instrumental and subjective assessment, to prevent upper band synthesis during speech pauses and background noise. Based on the WB PSD $\tilde{\mathbf{\Phi}}_\ell^{\mathrm{WB}}$, it outputs frame-wise hard-decision values

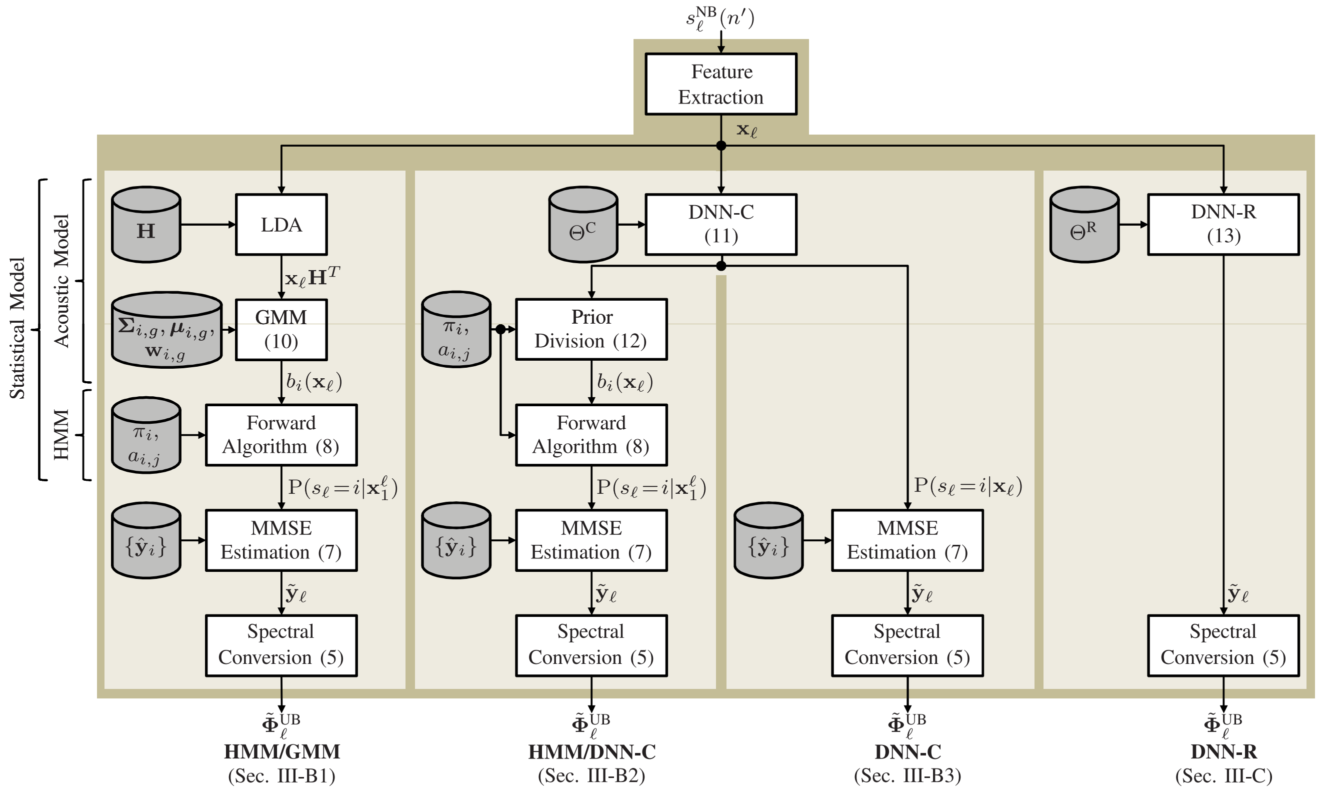

Fig. 2. Block diagram of *UB spectral envelope estimation*: The first three columns are classification-based UB cepstral estimation approaches, estimating probabilities of entries in a pretrained UB cepstral envelope codebook. All approaches convert into the spectral power domain to obtain the estimated spectral envelope. The last approach estimates the UB cepstrum $\tilde{\mathbf{y}}_\ell$ directly (regression-based) and, therefore, does not need a codebook. Equation numbers are given in ().

$VAD_\ell \in \{0,1\}$ for speech absence and speech presence, respectively. This output is multiplied against the residual signal and therefore controls the extension of the UB speech signal.

Spectral floor suppression (SFS) is a technique for controlling the synthesized energy in the UB for sounds with higher and lower UB energy separately. First, the ratio

$$R_\ell^{\mathrm{SFS}} = \frac{\frac{1}{|\mathcal{K}^{\mathrm{UB}}|} \cdot \sum\limits_{k\in\mathcal{K}^{\mathrm{UB}}} \left|\tilde{\Phi}_\ell^{\mathrm{WB}}(k)\right|^2}{\frac{1}{|\mathcal{K}^{\mathrm{NB}}|} \cdot \sum\limits_{k\in\mathcal{K}^{\mathrm{NB}}} \left|\tilde{\Phi}_\ell^{\mathrm{WB}}(k)\right|^2}, \tag{1}$$

is calculated, with $\mathcal{K}^{\mathrm{UB}}$ ($\mathcal{K}^{\mathrm{NB}}$) being the set of frequency bins representing the UB (NB) power spectrum, and $|\mathcal{K}|$ denoting the cardinality of the sets. Two attenuation factors are defined: d_{low} [dB] for the attenuation of frames in case of $R_\ell^{\mathrm{SFS}} = 0$, and d_{high} [dB] for frames with $R_\ell^{\mathrm{SFS}} \geq \theta^{\mathrm{SFS}}$, i.e., where the UB energy is relatively high. An empirically-found threshold $\theta^{\mathrm{SFS}} = 0.5$ enables a good separation of both cases.

The transition of the logarithmic attenuation factor for $0 < R_\ell^{\mathrm{SFS}} < \theta^{\mathrm{SFS}}$ is chosen to be linear:

$$d_\ell = \min\left\{\frac{d_{\mathrm{high}} - d_{\mathrm{low}}}{\theta^{\mathrm{SFS}}} \cdot R_\ell^{\mathrm{SFS}} + d_{\mathrm{low}}, d_{\mathrm{high}}\right\}, \tag{2}$$

with $d_{\mathrm{high}} = -3$ and $d_{\mathrm{low}} = -9$. Finally, the estimated UB speech signal is calculated as

$$\hat{s}^{\mathrm{UB}}(n) = \tilde{s}^{\mathrm{UB}}(n) \cdot 10^{\frac{d_\ell}{20\,\mathrm{dB}}}. \tag{3}$$

SFS can be used to individually attenuate or amplify the estimated UB for speech segments with high and low ratio R_ℓ^{SFS} and is therefore a powerful parameterization method to adapt the ABE algorithm to listeners' preferences. Accordingly, SFS is used later on in *all* investigated baseline and new approaches during *subjective* assessment.

Please note that neither VAD, nor SFS, have any impact on the UB spectral envelope estimation.

III. UB Spectral Envelope Estimation

We do not directly estimate the missing UB envelope PSD $\tilde{\boldsymbol{\Phi}}_\ell^{\mathrm{UB}}$, since it is highly redundant. Instead, we first estimate an UB spectral envelope in parametric form, i.e., a cepstral vector $\tilde{\mathbf{y}}_\ell$, which is then converted into the spectral domain.

In general, vector $\mathbf{y}_\ell$ is chosen of dimension $N_{\mathrm{env}} = 9$ [31]. The first element of $\mathbf{y}_\ell$ represents a logarithmic energy ratio

$$\mathrm{y}_\ell(0) = \frac{1}{\sqrt{2}} \cdot \ln\left(\frac{g_\ell^{\mathrm{UB}}}{g_\ell^{\mathrm{NB}}}\right), \tag{4}$$

with g_ℓ^{NB} and g_ℓ^{UB} being the prediction gain factors from LP analysis for the NB and UB PSD, respectively. The remaining coefficients $\mathrm{y}_\ell(1), \ldots, \mathrm{y}_\ell(N_{\mathrm{env}}-1)$ are linear (*not* Mel) cepstral coefficients describing the UB envelope.

UB spectral envelope estimation is conducted as depicted in Fig. 2. First, features from the input signal $s^{\mathrm{NB}}(n')$ are extracted (Section III-A). Based on feature vector $\mathbf{x}_\ell$, UB cepstral estimation is performed, which outputs an UB cepstrum $\tilde{\mathbf{y}}_\ell$. Section III-B will present three methods, which are based on classification of states related to UB codebook entries, while Section III-C focuses on a DNN-based statistical model which directly outputs $\tilde{\mathbf{y}}_\ell$.

Finally, using (4), the UB envelope-related cepstrum $\tilde{\mathbf{y}}_\ell$ is converted to the UB power spectral envelope

$$\begin{aligned}\tilde{\Phi}_\ell^{\text{UB}}(k) &= g_\ell^{\text{NB}} \cdot \exp\left(\sqrt{2}\cdot \tilde{\mathrm{y}}_\ell(0) + 2\mathrm{Re}\left\{\sum_{\nu=1}^{N_{\text{env}}} \tilde{\mathrm{y}}_\ell(\nu) e^{-j2\pi \frac{\nu k}{|\mathcal{K}^{\text{UB}}|}}\right\}\right)\\ &= g_\ell^{\text{NB}} \cdot \frac{\tilde{g}_\ell^{\text{UB}}}{\tilde{g}_\ell^{\text{NB}}} \cdot \exp\left(2\mathrm{Re}\left\{\sum_{\nu=1}^{N_{\text{env}}} \tilde{\mathrm{y}}_\ell(\nu) e^{-j2\pi \frac{\nu k}{|\mathcal{K}^{\text{UB}}|}}\right\}\right).\end{aligned} \tag{5}$$

During ABE processing, g_ℓ^{NB} is obtained on the input NB signal via LP analysis (Fig. 1) and then used to compensate the denominator of the estimated LP gain ratio $\tilde{g}_\ell^{\text{UB}}/\tilde{g}_\ell^{\text{NB}}$, taken from $\tilde{\mathrm{y}}_\ell(0)$, which is the estimated log-energy ratio (4).

A. Feature Extraction

Two different feature vector definitions are used in this work. They differ in the particular NB envelope representation, which is either based on autocorrelation function (ACF) coefficients [31], or on log-mel filter bank (Fbank) coefficients [32]. In both feature vector definitions, however, the following five features are included, which capture highly relevant characteristics of the input NB signal for the ABE task [12], [31], [13], [19], [16]: Zero-crossing rate and gradient index for voiced/unvoiced indication, normalized relative frame energy for detection of sudden changes in NB energy over frames, spectral centroid for fricative sound indication, and local kurtosis for plosive and vocal sounds indication. The static feature set $\bar{\mathbf{x}}_\ell$ is then obtained by appending all of these to either 10 ACF or 29 Fbank coefficients.

Furthermore, temporal dynamics are calculated using a symmetrical regression window centered at frame ℓ, with lookahead and look-back of one frame to ensure still a low delay as required for conversational telephone applications. The final feature vector is then composed as follows

$$\mathbf{x}_\ell = \left[\bar{\mathbf{x}}_\ell^T, \bar{\mathbf{x}}_{\ell+1}^T - \bar{\mathbf{x}}_{\ell-1}^T, \bar{\mathbf{x}}_{\ell+1}^T - 2\bar{\mathbf{x}}_\ell^T + \bar{\mathbf{x}}_{\ell-1}^T\right]^T, \tag{6}$$

with $()^T$ being the transpose.

The ACF-based feature definition has $(10+5)\times 3 = 45$ dimensions, while the Fbank-based feature definition is of dimension $(29+5)\times 3 = 102$.

B. Classification With and Without HMM

Classification-based UB spectral envelope estimation requires a codebook, which contains a set of pretrained cepstral envelope vectors $\{\hat{\mathbf{y}}_i\}$, with $i = 1, \ldots, N$, found via clustering of training data. During ABE processing, a statistical model obtains *a posteriori* probabilities $\mathrm{P}(s_\ell = i|\mathbf{x}_\ell)$, where state $s_\ell = i$ corresponds to codebook entry i.

Using the minimum mean square error (MMSE) estimation rule, the *a posteriori* probabilities $\mathrm{P}(s_\ell = i|\mathbf{x}_\ell)$ serve as weights for the codebook entries $\{\hat{\mathbf{y}}_i\}$, which leads to an estimated UB cepstral vector:

$$\tilde{\mathbf{y}}_\ell = \sum_{i=1}^{N} \mathrm{P}(s_\ell = i|\mathbf{x}_\ell) \cdot \hat{\mathbf{y}}_i. \tag{7}$$

The estimated UB cepstral vector is then converted to the spectral power domain via (5).

In this work, three statistical models for classification-based UB spectral envelope estimation are employed, depicted in Fig. 2 in the first three columns. The first two (**HMM/GMM** and **HMM/DNN-C**) are HMM-based approaches, while the third approach drops the HMM (**DNN-C**). HMM-based statistical models obtain *a posteriori* probabilities $\mathrm{P}(s_\ell = i|\mathbf{x}_1^\ell)$ via the forward algorithm (FA)[1] [33]. For the first frame, the forward variable is initialized by $\alpha_1(i) = \pi_i b_i(\mathbf{x}_1)$, with initial state probability $\pi_i := \mathrm{P}(s_i)$ and observation (emission) $b_i(\mathbf{x}_\ell) := p(\mathbf{x}_\ell|s_i)$. In this work, the observation is calculated via a GMM (Section III-B1) or a DNN (Section III-B2). For the subsequent frames, the forward variable is calculated as follows:

$$\alpha_{\ell+1}(j) = b_j(\mathbf{x}_{\ell+1}) \cdot \sum_{i=1}^{N} \alpha_\ell(i) \cdot a_{i,j}, \tag{8}$$

with $a_{i,j} := \mathrm{P}(s_{\ell+1} = j|s_\ell = i)$ being the transition probability from state i to state j. Finally, *a posteriori* probabilities are calculated by simple normalization

$$\mathrm{P}(s_\ell = i|\mathbf{x}_1^\ell) = \frac{\alpha_\ell(i)}{\sum\limits_{j=1}^{N} \alpha_\ell(j)}. \tag{9}$$

Since our DNN for classification will have a softmax output layer, *a posteriori* probabilities are available at its output and thus can also be used without application of the FA for weighting the codebook entries in (7). This case is denoted as **DNN-C** and is explained in Section III-B3.

*1) GMM-Based HMM (**HMM/GMM**):* In the left part of Fig. 2 the processing of a GMM-based HMM, denoted as **HMM/GMM**, is shown. It is based on a dimension-reduced feature vector $\mathbf{x}_\ell \mathbf{H}^T$, where $\mathbf{H}^T$ is the pretrained linear discriminant analysis (LDA) matrix, which lowers the dimension in case of ACF-based feature vectors (c.f. Section III-A) from $\dim(\mathbf{x}_\ell) = 45$ to $\dim(\mathbf{x}_\ell \mathbf{H}^T) = 9$.

Using GMMs, the observation probability is calculated as

$$b_i(\mathbf{x}_\ell) = \sum_{g=1}^{G} w_{i,g} \cdot \mathcal{N}\left(\mathbf{x}_\ell \mathbf{H}^T; \boldsymbol{\mu}_{i,g}, \boldsymbol{\Sigma}_{i,g}\right), \tag{10}$$

with $w_{i,g}$ being state-dependent weights for the G modes, $\boldsymbol{\mu}_{i,g}$ the mean vectors, $\boldsymbol{\Sigma}_{i,g}$ the diagonal covariance matrices, and the weighted sum of multivariate Gaussian probability density functions $\mathcal{N}(.)$ approximating the probabilistic distribution of the dimension-reduced feature vectors characterizing state i.

[1] Please note that due to the recursive calculation of the FA, actually all past feature vectors $\mathbf{x}_1^\ell = [\mathbf{x}_1, \ldots, \mathbf{x}_\ell]$ have an influence on the calculation of the current *a posteriori* probabilities.

*2) DNN-Based HMM (**HMM/DNN-C**):* In general, processing of a feature vector through a DNN provides the DNN output activity $\mathbf{f}(\mathbf{x}_\ell;\Theta)$ with elements $f_i(\mathbf{x}_\ell;\Theta)$ and Θ being the set of network parameters comprising weights and biases. The set of DNN parameters for classification purposes is denoted as Θ^{C}, so that we obtain

$$\mathrm{P}(s_\ell = i|\mathbf{x}_\ell) = f_i(\mathbf{x}_\ell;\Theta^{\mathrm{C}}). \qquad (11)$$

The approach denoted as **HMM/DNN-C** is presented in the second column of Fig. 2. The observation probability is calculated from the DNN output, i.e., *a posteriori* probabilities, by dividing out the state prior:

$$b_i(\mathbf{x}_\ell) = \frac{f_i(\mathbf{x}_\ell;\Theta^{\mathrm{C}})}{\pi_i}. \qquad (12)$$

*3) DNN Without HMM (**DNN-C**):* As depicted in the third column in Fig. 2, the estimated *a posteriori* probabilities (11) are directly used as weights for (7). Since this calculation is memoryless, $\mathrm{P}(s_\ell = i|\mathbf{x}_\ell)$ depends only on frames with indices $\ell-1, \ell, \ell+1$ according to the definition of $\mathbf{x}_\ell$ (6).

C. Regression Without HMM (*DNN-R*)

Finally, the DNNs can directly output an estimation of the UB cepstral vector $\tilde{\mathbf{y}}_\ell$ based on feature vector $\mathbf{x}_\ell$, when used with the set of parameters Θ^{R}:

$$\tilde{\mathbf{y}}_\ell = \mathbf{f}(\mathbf{x}_\ell;\Theta^{\mathrm{R}}). \qquad (13)$$

The same limited temporal context is used here as for variant **DNN-C**. This variant is shown in the last column of Fig. 2. Using regression-based statistical models eliminates the need for a codebook $\{\hat{\mathbf{y}}_i\}$.

IV. Experimental Setup, Training, and Metrics

The experimental setup is shown in Fig. 3. A detailed overview is given in the following subsections.

A. Speech Databases and Preprocessing

Speech data used in this work is taken from the TIMIT database [34], Speechdat-Car US (SDC) database [35], and NTT database [36], as depicted in the upper part of Fig. 3. The use of a different database in test could be considered as an extra challenge to prove generalization capabilities of the choices made on the validation data. To prevent overlap w.r.t. sentences, the so-called "SA" sentences from the TIMIT database were ignored. From the SDC database, which was recorded in an automotive environment, only phonetically rich sentences are used. For testing, German and American English parts of the NTT database are used. We use the "TRAIN" part of the TIMIT as training data and the "TEST" part as validation data. Also, the SDC database is split into two sets (no speaker overlap), using the respective ratios from the TIMIT corpus, and is appended to form the training and validation sets. Accordingly, the training and validation sets are both a mixture of the TIMIT and SDC databases. As a result, the training set consists out of 5.8 h, the validation set out of 2.4 h, and the test set out of 0.4 h of speech material.

Preprocessing of the speech data as shown in Fig. 3 is done according to [5]. NB preprocessing uses the WB speech signals from the databases, conducting MSIN filtering [37], followed by decimation to 8 kHz, 16 to 13 bit conversion, adaptive multirate (AMR) coding at 12.2 kbps and immediate decoding [38], and final 16 to 13 bit conversion. The resulting NB condition is input to the ABE framework. WB preprocessing aims at simulating the capabilities of a WB-capable phone, performed by P.341-filtering [37] of the available WB speech data. Delay introduced in this preprocessing scheme was compensated for.

B. Selective Linear Prediction Analysis

SLP analysis is a technique for calculating a parametric representation of the UB spectral envelope in the cepstral domain, when WB speech is available [39], [31]. First, the preprocessed WB speech signal is transformed into the frequency domain using the DFT. The squared absolute of the short-term spectrum will lead to the WB PSD $\Phi_\ell^{\mathrm{WB}}(k)$. LP analysis is carried out only for the UB DFT bins $k \in \mathcal{K}^{\mathrm{UB}}$, leading to filter coefficients a_ℓ^{UB} and prediction gain g_ℓ^{UB}. Via recursive calculation [6], a vector of cepstral coefficients $\mathbf{c}_\ell$ is obtained. From the respective NB preprocessed speech, g_ℓ^{NB} is calculated via LP analysis. Finally, the UB cepstral vector is composed as (cf. (4)):

$$\mathbf{y}_\ell = \left[\frac{1}{\sqrt{2}} \cdot \ln\left(\frac{g_\ell^{\mathrm{UB}}}{g_\ell^{\mathrm{NB}}}\right), \mathbf{c}_\ell^T\right]^T. \qquad (14)$$

C. Codebook Design

The employed codebook design is based on Bauer's approach [13] and requires frame-wise transcription of the training data by phonetic labels[2] $\varphi_\ell \in \{0,1\}$, where $\varphi_\ell = 1$ stands for a speech frame containing an /s/ or /z/ sound and $\varphi_\ell = 0$ for all other sounds. By means of the preprocessed NB and WB speech signals, SLP analysis outputs an UB cepstral vector $\mathbf{y}_\ell$ for every frame $\ell \in \mathcal{L}$, with $\mathcal{L}$ being the set of all frames occurring in the training data. The set of frames $\mathcal{L}$ is split into two subsets of frames: $\mathcal{L}_{\overline{/s/}} = \{\ell|\varphi_\ell = 0 \wedge \ell \in \mathcal{L}\}$ and $\mathcal{L}_{/s/} = \{\ell|\varphi_\ell = 1 \wedge \ell \in \mathcal{L}\}$.

For both of the subsets, an individual (partial) codebook is calculated. An isolated consideration of /s/ and /z/ sounds, i.e., extra codebook entries for these sounds, is justified by the fact, that data-driven Linde-Buzo-Gray (LBG) clustering of all frames $\ell \in \mathcal{L}$ would lead to high reconstruction errors of these fricative sounds [40].

Via purely data-driven LBG clustering of subset $\mathcal{L}_{\overline{/s/}}$, codebook entries $\{\hat{\mathbf{y}}_i\}, i = 1, \ldots, 16$ are found. Codebook entries for /s/ and /z/ sounds are found in two steps: An LBG derives a prototype codebook with 64 classes from subset $\mathcal{L}_{/s/}$. Then 8 entries are chosen, which have the highest Euclidean distance to the mean vector of all vectors from subset $\mathcal{L}_{/s/}$. The so-found entries are stored in the codebook at indices $i = 17, \ldots, N = 24$ and thereby enable a good representation of /s/ and /z/ sounds in the codebook.

[2] Phonetic labels for the SDC database have been obtained beforehand using a phoneme recognizer. Note that the TIMIT database already provides high-quality phonetic labels.

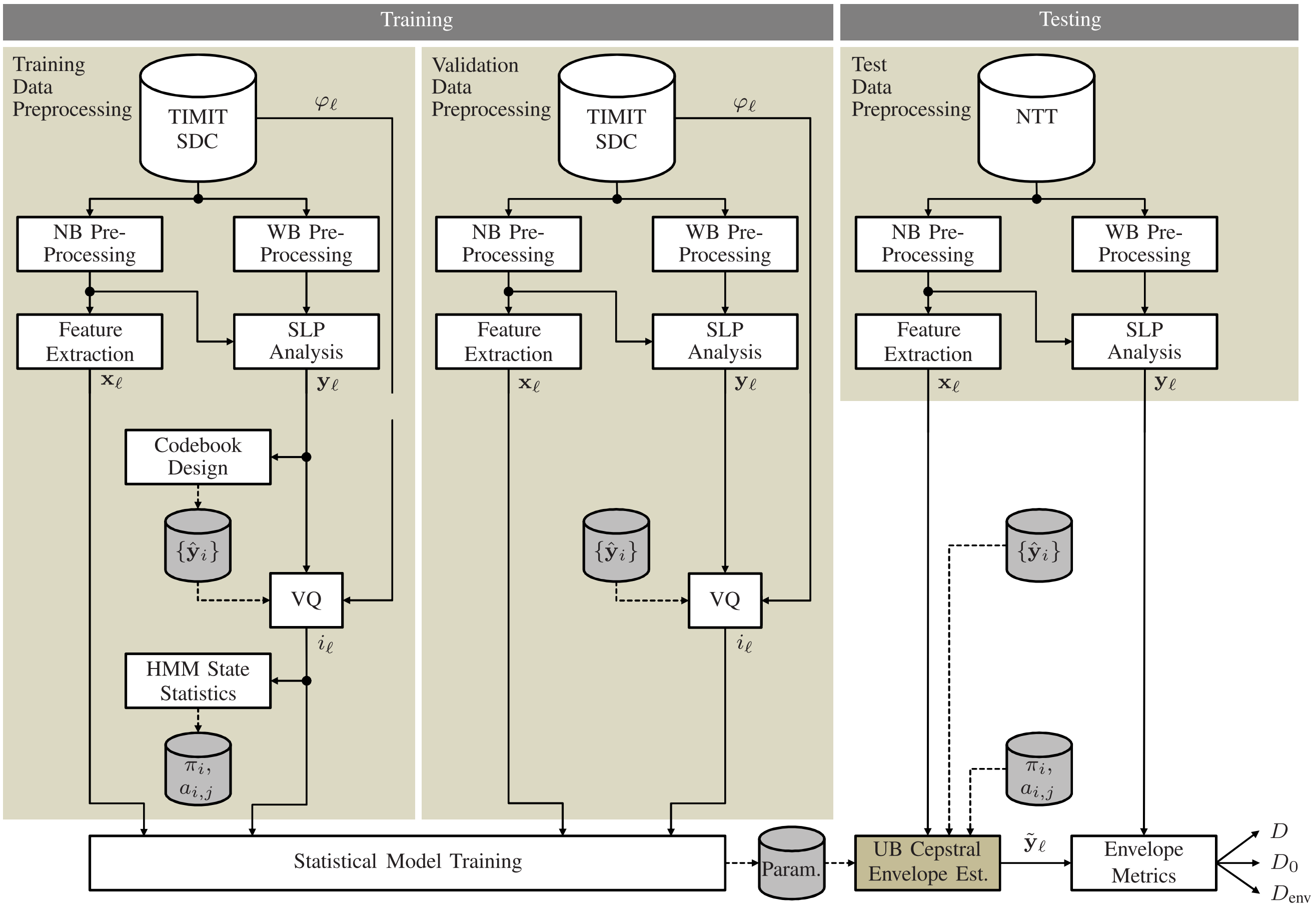

Fig. 3. Block diagram of the *experimental setup*. On the left and center: Generation of input and target pairs on training and validation data for statistical model training. On the right: Generation of input and target pairs on the test set for evaluation.

Note that this is a codebook design specifically useful for ABE [13]. If, however, a different type of codebook were used, the subsequent analysis and results are still expected to be valid w.r.t. relative performance of the conditions.

D. Statistical Model Training

By means of the codebook (c.f. Section IV-C), all UB cepstral envelope vectors $\mathbf{y}_\ell$, calculated via SLP analysis, are subject to vector quantization (VQ), assigning a state $i_\ell \in \{1, \ldots, N\}$ to each frame. This quantization is done in a supervised fashion: For $\varphi_\ell = 0$ only the first 16 codebook entries are used in the VQ and for $\varphi_\ell = 1$ only the last 8 entries are used.

1) HMM: State probabilities π_i are calculated using the normalized occurrences of each state i in the set $\{i_\ell | \ell \in \mathcal{L}\}$. In the same way, the state transition probabilities $a_{i,j}$ are found via simple counting and normalizing the state transitions found on the training data.

2) GMM: For each of the states $i \in \{1, \ldots, N\}$ defined by the codebook entries, a GMM with G modes is trained. Weights $w_{i,g}$, mean vectors $\boldsymbol{\mu}_{i,g}$, and diagonal covariance matrices $\boldsymbol{\Sigma}_{i,g}$ are found via the expectation-maximization (EM) algorithm [41], using feature vectors $\mathbf{x}_\ell$ and the corresponding quantized state indices i_ℓ. To reduce the amount of parameters to train, a dimension reduction via LDA is performed before the EM algorithm derives the parameters for the GMMs. The LDA transformation matrix $\mathbf{H}$ is found on data derived from the training set. Please note that the EM algorithm does not make use of the validation data.

3) DNN: Given the feed-forward DNN topology by a number of hidden layers N_{HL} and a number of units N_{U} per hidden layer, the weights and biases, jointly denoted as Θ, are initialized using randomly picked values from a uniform distribution with zero mean and standard deviation of $u^{-1/2}$, where u is the number of incoming connections of the respective unit [42]. Instead of initializing the parameters randomly, we also evaluate a restricted Boltzmann machine (RBM) pretraining of Θ using the training data according to [43]. In general, sigmoid activation functions are used in the entire network, except for classification-based DNNs, where a softmax layer forms the output. In [44] rectified linear units (ReLUs) have been compared to sigmoid activation functions for DNNs for automatic speech recognition and showed high potential for better estimation performance. Sigmoid activation functions are likely to cause the vanishing gradient problem [45], which results in slower convergence of the network's parameters during training and might even be responsible for finding only a poor local minimum. Please note that we conduct RBM pretraining only for sigmoid-driven DNNs.

The parameter set Θ is trained via the backpropagation (BP) algorithm [41]: Input data is passed through the network and compared to the desired output (target). Based on the error, parameters Θ are updated. In this work, we organize the available training data in minibatches $b \in \{1, \ldots, B\}$, i.e., subsets of a complete epoch's frames $\ell \in \mathcal{L}$, each containing $L = 128$ frames indexed by $\lambda \in \{1, \ldots, L\}$. After each minibatch, an update step is performed based on the average error of minibatch b:

$$J_b(\Theta_b) = \frac{1}{L} \sum_{\lambda=1}^{L} J_{b,\lambda}(\Theta_b). \tag{15}$$

For the classification-based application of the DNN, the negative log-likelihood (NLL) error function is used:

$$J_{b,\lambda}(\Theta_b^{\mathrm{C}}) = -\ln\left(f_{i=i_\lambda}(\mathbf{x}_{b,\lambda}; \Theta_b^{\mathrm{C}})\right), \tag{16}$$

with i_λ being the index of the target state for frame λ in minibatch b. For regression-based applications, the mean squared error (MSE) function is used:

$$J_{b,\lambda}(\Theta_b^{\mathrm{R}}) = \frac{1}{2} \left|\left|\mathbf{f}(\mathbf{x}_{b,\lambda}; \Theta_b^{\mathrm{R}}) - \mathbf{y}_{b,\lambda}\right|\right|^2. \tag{17}$$

One major difference between the two error functions is that the NLL error function considers only the error at the target state, while for MSE all outputs influence the error which is then backpropagated. Parameter set Θ is then updated following

$$\Theta_{b+1} = \Theta_b - \eta \nabla J_b(\Theta_b), \tag{18}$$

with $\eta = 0.1$ being the learning rate and ∇ being the gradient operation. We also apply momentum and L2-regularization during parameter update steps [41]. After all minibatches have been processed, i.e., after each epoch, stopping criteria are checked. On the one hand, we set a maximum number of 50 epochs for the parameter training. On the other hand, we check the error on the validation set using the final parameter set Θ at the end of an epoch and compare it to the error calculated in the previous epoch. Since an increased error indicates overfitting of the network to the training data, we then start the next epoch's learning step using the parameter set from the previous epoch but with only half of the learning rate. If a minimum learning rate of $\eta_{\min} = 0.0001$ is reached, we stop the training.

We also investigate dropout as technique for generalizing the DNN training [46]. Using dropout will only train a subset of the DNN parameter set Θ at a time. Per epoch only 20% of the input units and 50% of the units in the hidden layers will be considered during the backpropagation steps. Which units are dropped will be decided at the beginning of an epoch.

Mean and variance normalization of feature vectors from training, validation, and test data is conducted using statistics found on the training data, intentionally providing conservative results on the test data due to the acoustic differences of the datasets.

TABLE I
RESULTS FOR **HMM/GMM** VARYING THE NUMBER OF MODES G PER GMM

# Modes	Validation Set				Test Set			
G	D	D_0	D_{env}	Acc	D	D_0	D_{env}	Acc
4	8.12	6.29	4.12	23.22	**9.98**	**7.79**	**5.32**	**22.48**
8	7.95	6.09	4.11	23.37	10.31	8.11	5.45	18.57
16	7.90	6.06	**4.10**	23.93	10.11	**7.79**	5.53	17.24
32	**7.86**	**6.01**	**4.10**	**24.33**	10.33	8.04	5.50	17.01

Cepstral distances are given in [dB] and accuracy in %.

E. Metrics for UB Spectral Envelope Quality

To instrumentally evaluate the estimated UB spectral envelope, a cepstral distance is calculated [47]:

$$D = 10\sqrt{2} \cdot \log_{10}(e) \sqrt{\sum_{\nu=0}^{N_{\text{env}}} \left(\mathrm{y}(\nu) - \tilde{\mathrm{y}}(\nu)\right)^2} \text{ [dB]}. \tag{19}$$

Furthermore, a modified cepstral distance focusing only on the envelope-relevant estimated UB cepstral coefficients, calculated as

$$D_{\text{env}} = 10\sqrt{2} \cdot \log_{10}(e) \sqrt{\sum_{\nu=1}^{N_{\text{env}}} \left(\mathrm{y}(\nu) - \tilde{\mathrm{y}}(\nu)\right)^2} \text{ [dB]} \tag{20}$$

is employed for evaluation. Since the value of $\tilde{\mathrm{y}}(0)$ plays such an important role in ABE approaches, we will also report separately on

$$D_0 = 10\sqrt{2} \cdot \log_{10}(e) \cdot \left|\mathrm{y}(0) - \tilde{\mathrm{y}}(0)\right| \text{ [dB]}. \tag{21}$$

These three distances are calculated per frame. The reported metrics have subsequently been averaged over frames of each file and finally over all files of the respective data set.

For classification-based experiments, the accuracy [%] (Acc) of correctly predicted state indices i in the maximum a posteriori fashion will be given.

V. EXPERIMENTS, EVALUATION, AND DISCUSSION

First, in Section V-A, we will evaluate **HMM/GMM** as statistical model for UB spectral envelope estimation. Afterwards, we will move on to the DNN-driven statistical models **HMM/DNN-C**, **DNN-C**, and **DNN-R** and will investigate various training parameters and configurations in Section V-C for a given network topology, which is found in a preliminary experiment in Section V-B.

A. Baseline: ***HMM/GMM***

In this baseline experiment UB spectral envelope estimation is conducted via the HMM/GMM-based classification scheme as presented in Section III-B1 [13]. The number of modes G used per GMM is investigated. Results are presented in Table I.

On the validation data we observe monotonously decreasing cepstral distances for a higher number of modes G, while state accuracy is increasing. The lowest cepstral distances on the test data, however, were achieved for the smallest number

TABLE II
RESULTS FOR **HMM/DNN-C** FOR $N_{\mathrm{HL}} = 1, \ldots, 6$ HIDDEN LAYERS AND THE NUMBER OF UNITS $N_{\mathrm{U}} = 256, 512, 1024$ PER HL

Topology		Validation Set				Test Set			
N_{HL}	N_{U}	D	D_0	D_{env}	Acc	D	D_0	D_{env}	Acc
1	256	7.32	5.48	**3.96**	29.56	9.69	7.22	5.54	21.75
	512	7.33	5.49	**3.96**	29.74	9.66	7.19	5.53	22.46
	1024	7.32	5.48	**3.96**	29.73	9.63	7.14	**5.52**	22.73
2	256	7.31	5.47	3.97	30.96	9.85	7.36	5.65	20.06
	512	7.32	5.48	**3.96**	31.11	9.97	7.54	5.60	20.43
	1024	7.30	5.46	3.97	31.04	10.12	7.70	5.67	17.75
3	256	7.30	5.46	3.97	31.17	9.72	7.21	5.61	21.52
	512	7.31	5.48	**3.96**	31.29	9.80	7.35	5.55	22.66
	1024	7.31	5.48	**3.96**	31.19	9.87	7.43	5.57	21.34
4	256	7.27	5.42	**3.96**	31.46	9.75	7.23	5.65	20.00
	512	**7.26**	**5.42**	**3.96**	**31.48**	**9.57**	**7.10**	5.53	**23.88**
	1024	7.29	5.45	**3.96**	31.31	9.77	7.32	5.58	21.70
5	256	7.27	5.42	**3.96**	31.24	9.60	7.12	5.56	22.82
	512	7.29	5.45	**3.96**	31.22	9.70	7.24	5.58	21.29
	1024	7.28	5.43	3.97	31.15	9.79	7.28	5.65	19.81
6	256	7.51	5.65	4.06	26.40	9.66	7.15	5.64	18.38
	512	7.43	5.58	4.04	26.69	9.65	7.13	5.65	17.89
	1024	7.45	5.53	4.13	23.21	9.91	7.35	5.78	13.60

Cepstral distances are given in [dB] and accuracy in %. The number of parameters to train is *not* equal.

of modes ($G = 4$), resulting from the different databases being used in training/validation and test, and accordingly indicating overfitting of the GMMs on the training data.

Due to the best performance on validation data, we select the $G = 32$ case as **HMM/GMM** baseline, but we will also keep an eye on the $G = 4$ case.

B. Investigation of DNN Topology

We aim at evaluating all training parameters and configurations as presented in the former sections. The root of all DNN-related experiments is the network topology. In [28] we found that the number of hidden layers N_{HL} and the number of units N_{U} per hidden layer have minor influence on the UB spectral envelope estimation. Accordingly, in this preliminary experiment, we will find the best performing topology parameter set $(N_{\mathrm{HL}}, N_{\mathrm{U}})$ w.r.t. to lowest cepstral distances on validation data and then use *this particular* topology setting for all other investigations.[3]

We employ **HMM/DNN-C** with default training parameters: Sigmoid activation, ACF-based feature vector definition, no dropout, and no pretraining of network parameters. The results are shown in Table II. Most of the presented $(N_{\mathrm{HL}}, N_{\mathrm{U}})$ choices for the DNN outperform all of the GMM-based ABE cases in Table I. Considering the performance for a comparable number of model parameters, **HMM/GMM** with $G = 32$ (15.2 K parameters) and **HMM/DNN-C** with $N_{\mathrm{HL}} = 1$ and $N_{\mathrm{U}} = 256$ (17.9 K parameters), the superiority of DNNs over GMMs become obvious. However, we observe that the pure envelope-related UB cepstral distance D_{env} on the *test* set slightly degraded with the DNNs, asking for regularization methods in subsequent experiments.

As expected, among the different $(N_{\mathrm{HL}}, N_{\mathrm{U}})$ settings differences in performance are rather small, even though the number of parameters to train vary quite a lot over the different conditions. The lowest cepstral distances on the validation data were achieved by the DNN with $N_{\mathrm{HL}} = 4$ hidden layers and $N_{\mathrm{U}} = 512$ units per hidden layer. Further experiments will be based on this topology.

C. Evaluation and Discussion of DNN Training Parameters

Given the DNN topology with $N_{\mathrm{HL}} = 4$ and $N_{\mathrm{U}} = 512$ as found in Section V-B, we evaluated all statistical models for all of the so-far presented training options, namely NB envelope representation (as part of the feature vector), unit type, pretraining, and dropout. The results are presented in Table III. Note that the first row in Table III corresponds to the winning topology in Table II.

For *all* of the three DNN-driven statistical models the best performing DNN configurations on validation data w.r.t. cepstral distances (and accuracy) are the Fbank-based feature definition with pretrained sigmoid units, ReLUs (second best), and sigmoid units without pretraining (third best).

To investigate the influence of a priori knowledge, i.e., the consideration of initial state probabilities and state transition probabilities as given by the *HMM* during UB cepstral estimation, we compare **HMM/DNN-C** to **DNN-C**. First of all, we notice a small increase in the classification accuracy for **DNN-C** on the validation data. For the cepstral distance D a clear improvement arises for **DNN-C** (7.03 → 6.62 dB), which is mainly caused by an improved D_0, i.e., a better UB energy estimation. A view on the results for the test data reveals a different picture of the best validation data approach: Accuracy and cepstral distances are (slightly) worse than those of **HMM/DNN-C**. Obviously, **DNN-C** suffers a little more from overfitting to the training data than **HMM/DNN-C**. An explanation might be that the a priori knowledge from the HMM compensates for overfitting of the acoustic model.

Pretraining of weights and biases for the DNN with sigmoid activation function was found to be beneficial in quite some cases: When used with the ACF-based feature definition a surprisingly good D_0 performance on the test set was achieved for all statistical models. On the other hand, pretraining on the Fbank-based feature definition *always* improved cepstral distances and accuracy both on validation and test data. Given the statistical model **DNN-C** with ACF-based feature definition and sigmoid units, the DNN training stops after 50 epochs (stopping criterion satisfied) and reaches a final accuracy of 39.49% on the training data. The learning rate is first halved after epoch 24. In contrast, the ReLU-based DNN training stops already after epoch 31, when the minimum learning rate is reached. The learning rate is first halved after epoch 2. The final accuracy on the training data is 40.26%. Consequently, we can support the earlier findings that ReLUs enable faster learning and lead to a higher model accuracy on the training data. Generalization

[3] Instead of using the results from [28], we have to repeat this experiment, since the underlying datasets differ from this previous investigation.

TABLE III
ALL **EXPERIMENTAL RESULTS** FOR THE DNN-BASED APPROACHES WITH $N_{\mathrm{HL}} = 4$ HIDDEN LAYERS AND $N_{\mathrm{U}} = 512$ UNITS PER LAYER

Statistical Model	NB Envelope Representation	Unit Type	Dropout	Pre-training	Validation Set				Test Set			
					D	D_0	D_{env}	Acc	D	D_0	D_{env}	Acc
HMM/DNN-C	ACF	Sigmoid	no	no	7.26	5.42	3.96	31.48	9.57	7.10	5.53	23.88
			yes	no	8.17	6.41	4.10	19.52	9.29	**6.48**	5.77	14.28
			no	yes	7.27	5.42	3.97	31.30	9.61	7.05	5.63	21.53
		ReLU	no	no	7.29	5.45	3.96	31.37	9.91	7.40	5.64	20.23
			yes	no	8.22	6.48	4.09	18.50	9.35	6.56	5.78	14.55
	Fbank	Sigmoid	no	no	7.06	5.23	**3.87**	34.90	9.31	7.09	5.22	28.56
			yes	no	7.44	5.64	3.96	25.96	9.87	7.70	5.39	20.75
			no	yes	**7.03**	**5.20**	**3.87**	**35.11**	**9.22**	6.99	**5.22**	**29.17**
		ReLU	no	no	**7.03**	**5.20**	**3.87**	34.97	9.31	7.06	5.25	28.24
			yes	no	7.66	5.89	3.97	24.29	9.92	7.73	5.36	19.73
DNN-C	ACF	Sigmoid	no	no	6.82	4.93	3.91	32.04	9.63	7.18	5.57	21.40
			yes	no	7.36	5.48	4.07	21.55	**9.20**	**6.44**	5.75	10.62
			no	yes	6.81	4.91	3.92	31.90	9.60	7.06	5.64	18.98
		ReLU	no	no	6.85	4.97	3.92	31.97	9.93	7.45	5.66	17.72
			yes	no	7.46	5.58	4.07	20.67	9.23	6.47	5.75	11.29
			no	no	6.65	4.79	**3.83**	35.66	9.41	7.20	5.32	25.79
	Fbank	Sigmoid	yes	no	6.85	4.94	3.95	28.32	9.90	7.66	5.53	17.31
			no	yes	**6.62**	**4.75**	**3.83**	**35.92**	9.31	7.08	**5.30**	**26.55**
		ReLU	no	no	6.64	4.77	**3.83**	35.76	9.40	7.15	5.34	25.46
			yes	no	7.01	5.12	3.95	26.71	9.93	7.67	5.53	15.91
DNN-R	ACF	Sigmoid	no	no	6.59	4.73	3.84	N/A	9.55	7.14	5.51	N/A
			yes	no	7.25	5.40	3.99	N/A	9.23	**6.70**	5.59	N/A
			no	yes	6.57	4.71	3.84	N/A	9.43	7.00	5.50	N/A
		ReLU	no	no	6.58	4.74	3.81	N/A	9.68	7.26	5.55	N/A
			yes	no	7.55	5.72	4.02	N/A	9.42	6.81	5.68	N/A
	Fbank	Sigmoid	no	no	6.41	4.59	3.72	N/A	9.39	7.30	5.16	N/A
			yes	no	6.83	4.97	3.89	N/A	9.93	7.81	5.41	N/A
			no	yes	**6.35**	**4.56**	**3.70**	N/A	**9.15**	7.10	**5.05**	N/A
		ReLU	no	no	6.36	**4.56**	3.71	N/A	9.37	7.24	5.19	N/A
			yes	no	7.13	5.29	3.88	N/A	10.01	7.84	5.41	N/A

CEPSTRAL DISTANCES are given in [dB] and **ACCURACY** in %. Bold numbers mark the best results for each of the statistical models. The first row in this table equals the lowest cepstral distance in Table II.

of the model on unseen data, however, suffers as proven by slightly higher cepstral distances on the validation data and much worse performance on test data compared to sigmoid-based units. Furthermore, for all DNN-based statistical models, ReLUs never lead to lower cepstral distances than pretrained sigmoid units, however, ReLUs are always close and therefore comparable. Considering the additional time required for conducting the RBM pretraining, using ReLUs might be a good practical alternative.

Regarding the use of *dropout*, we found that performance improves neither on the training nor on the validation data. A gain in performance could only be shown on the test data and only if the ACF-based feature definition is employed. For all experiments with Fbank-based feature definitions the performance on all of the three data sets was significantly decreased by dropout. This effect might be explained by the fact, that the Fbank-based feature definition contains more relevant and *non*-relevant information which needs to be learned and therefore implicitly regularizes the parameter update steps during BP. As a result, the overall regularization including dropout might be too strong and therefore interferes with finding an optimum training state during BP. By using the ACF-based feature definition, however, the overall best estimation of $\mathrm{y}(0)$ with $D_0 = 6.44$ dB on the test data was achieved by **DNN-C** with sigmoid units and the use of dropout.

In general, the Fbank-based feature definition leads to higher *state accuracies* than the ACF-based feature definition, on both the validation and test data. The highest state accuracies were obtained by DNN-based statistical models with Fbank-based feature definition and pretrained sigmoid units, where **DNN-C** was best on validation data and **HMM/DNN-C** was best on test data. A priori knowledge exploited by the HMM seems to be beneficial for UB spectral envelope estimation on unseen data sets. Comparing **HMM/DNN-C**, Fbank, pretrained sigmoid units with **DNN-C**, Fbank, and ReLUs with dropout on the validation data, we obtain similar cepstral distances $D = 7.03$ dB and $D = 7.01$ dB, respectively, but an absolute difference in accuracy of 8.4%. This shows that higher state accuracy does not necessarily correlate with lower cepstral distances.

Furthermore, the Fbank-based feature definition leads to the lowest cepstral distances on validation data for all DNN-driven statistical models. The overall lowest cepstral distances on validation data were achieved by **DNN-R**, Fbank, and pretrained sigmoid units. Keeping in mind that training and validation data stem from the same data source and that (a) the error function of regression-based DNN training enforces a stronger fit of the model to the training data than the NLL criterion (c.f. Section IV-D3), (b) the Fbank-based feature definition ($29 + 5$ coefficients) contains more information than the ACF-based feature definition ($10 + 5$ coefficients) and thus the potential for overfitting to characteristic aspects of the training data is high, and (c) RBM pretraining lets the network learn the training data better (instead of no pretraining), this particular configuration is not surprising for obtaining the best results on the validation data.

As mentioned before, the overall lowest D_0 on test data, however, was achieved by **DNN-C**, ACF-based feature definition, sigmoid units, and dropout. Due to the fact that test data is not from the same database as training and validation data, the positive effect of regularization methods becomes visible: DNN-C is a statistical model using a precalculated codebook, which can be seen as a priori knowledge or regularization, since the DNN does not need to learn how to generate cepstral vectors, but only how to separate them. Our ACF-based feature definition is a coarser representation than Fbank-based features of the underlying NB signal and therefore prevents the statistical model from overfitting. Furthermore, sigmoid units do not adapt as quickly as ReLUs and along with dropout a strong regularization scheme is applied.

Comparing **DNN-C** and **DNN-R** experiments with the same training parameters, the higher performance of the regression-based approach on the validation data can be traced back to both, a better UB cepstral envelope estimation and an improved UB energy estimation. For the test data, however, only improvements for the UB cepstral envelope could be shown for **DNN-R**, while for UB energy estimation an inconsistent picture results. Even though the MMSE estimation of classification-based statistical models enables a higher variety of possible $\tilde{\mathbf{y}}$ at the output of UB spectral estimation, the codebook is a bottleneck, since it consists only of a limited number of quantized UB cepstral vectors. W.r.t. the test data, which is derived from a different speech database than the codebook, the envelopes contained in the codebook might not be a good fit. The regression-based DNN approach, however, seems to have learned more statistical dependencies between the NB and the UB and is able to better adapt the estimation process to the underlying NB speech signals.

Concluding on the experiments, the best parametrization of the statistical models is the Fbank-based feature definition along with pretrained sigmoid units. Using this configuration, **DNN-R** led to the lowest cepstral distances on the validation data and is therefore considered as winner condition and consequently evaluated w.r.t. speech quality in the following section. Compared to the **HMM/GMM** baseline with $G = 32$ modes, the cepstral distance on the test set was improved using the best **DNN-R** approach by 10.33 dB − 9.15 dB = 1.18 dB.

TABLE IV
INSTRUMENTAL SPEECH QUALITY ASSESSMENT: MOS-LQO VALUES OBTAINED BY WIDEBAND PESQ FOR ALL DNN-BASED STATISTICAL MODELS USING THE FBANK-BASED FEATURE DEFINITION AND PRETRAINED SIGMOID UNITS

Statistical Model	Validation Set MOS-LQO	Test Set MOS-LQO
HMM/GMM	2.67	2.57
HMM/DNN-C	2.95	2.71
DNN-C	3.10	2.79
DNN-R	**3.27**	**2.80**
Oracle	3.45	3.23

The DNN topology is set to $N_{\mathrm{HL}} = 4$ and $N_{\mathrm{U}} = 512$. Additionally, results for the **HMM/GMM** baseline with $G = 32$ are shown. The Oracle experiment skips UB cepstral estimation and uses the cepstral vectors found via SLP analysis on the WB data of the respective dataset.

Finally, we simulated a concatenation of the Fbank features and the ACF features in the overall best setting (Fbank+ACF, sigmoid, no dropout, but pretraining). Apart from very bad subjective speech quality, both D and D_0 degrade severely on the test data ($D = 11.44$ dB, $D_0 = 9.83$ dB), while $D_{\mathrm{env}} = 4.96$ dB remains roughly the same. We explain this by the higher number of weights to be trained on the same limited training material, and therefore do not follow this path further.

VI. Assessment of Speech Quality

In this section, speech quality of artificially-extended speech signals is assessed. In Section VI-A we employ wideband (WB)-PESQ to instrumentally evaluate all statistical models using the respective parameters and training configuaration which led to the lowest cepstral distances on the validation data. Subsequently, in Section VI-B, we conduct a comparison category rating (CCR) test to also evaluate speech quality subjectively.

A. Instrumental Assessment

For instrumentally assessing the speech quality of the different UB spectral estimation techniques, we employ WB-PESQ [48], which outputs mean opinion score estimates, called MOS listening quality objective, MOS-LQO. Additionally, we evaluate an oracle experiment, which uses $\mathbf{y}_\ell$ calculated via SLP analysis of the WB speech signal. The results based on the same data as underlying Table III are presented in Table IV.

In line with the cepstral distance results of Table III, WB-PESQ confirms the best ABE speech quality for the **DNN-R** model. On the validation set **DNN-R** outperforms the **HMM/GMM** baseline by an impressive 0.6 MOS-LQO points improvement, just 0.18 points below the oracle. On the acoustically completely different test data **DNN-R** still achieved 0.23 MOS-LQO points improvement vs. **HMM/GMM**[4], with a 0.43

[4] Note that the **HMM/GMM** approach with best performance on the test set ($G = 4$) reached an MOS-LQO of only 2.53 points.

points gap towards the oracle. In general, the same rank order as for the presented cepstral distances in Table III is predicted by WB-PESQ.

In several publications, the comparison of *different* ABE approaches using WB-PESQ was found to malfunction when compared to ground truth subjective listening tests [49], [50]. However, here we only employ WB-PESQ to compare different variants of the *same* ABE approach to each other. Much better suited for instrumentally assessing speech quality of ABE approaches would be the QABE approach [51], however, this instrumental measure was trained on data processed by our HMM/GMM ABE approach and therefore reporting results calculated by QABE would not be scientifically sound.

B. Subjective Assessment

To also evaluate speech quality subjectively, we conducted a semi-formal CCR test, following ITU-T P.800 [52, Annex E]. In a CCR test, two conditions are compared to each other at once and rated on the comparison MOS (CMOS) scale from -3 (much worse) to +3 (much better) in steps of 1. Ten male and two female German listeners participated, who stated not to suffer from any hearing impairment. The subjects were compensated for their participation with a service charge.

The speech data presented in the CCR test is taken from the German part of the NTT database. Four conditions were derived from the speech data: One NB- and one WB-coded condition and two ABE approaches. First, the AMR condition is obtained as described in Section IV-A. The two ABE conditions are based on the AMR processed speech, where one employs **HMM/GMM** and **DNN-R** as statistical models, respectively. The **HMM/GMM** baseline uses $G = 32$ modes, since this configuration led to the best results on the validation data. The best DNN-based condition w.r.t. lowest cepstral distances on validation data is **DNN-R** with Fbank-based feature definition and pretrained sigmoid units and therefore used as third condition in the CCR test. The fourth condition represents coded WB speech, i.e., employing the WB preprocessing steps as described in Section IV-A with subsequent coding and decoding using AMR-WB operating at 12.65 kbit/s [53]. All files were bandpass-filtered to a frequency range of $0.2 \ldots 7$ kHz (similar to [19], [16]), considering the receiving sensitivity mask in [54]. Finally, the speech signals are scaled to an active speech level of -26 dBov [55] and converted to 48 kHz sampling rate. These four conditions require six comparisons leading to six CCR conditions in the subjective listening test.

The speech signals were presented in diotic fashion to the test subjects via two standard PCs with external `RME Fireface 400` sound cards using `AKG K-271 MKII` headphones. In a familiarization phase, test subjects listened to 12 sample pairs, containing all test conditions and were asked to choose a comfortable volume level. In the actual test, each test subject had to judge 36 file pairs in both sample orders, i.e., 72 sample pairs in total. Participants were allowed to listen to the samples more than once. Test subjects were equally assigned to one of two disjoint sets of randomized test files balanced over CCR conditions and speaker utterances.

TABLE V
SUBJECTIVE SPEECH QUALITY ASSESSMENT: RESULTS FROM A CCR TEST, EVALUATING THE HMM/GMM BASELINE AND THE BEST DNN-BASED ABE APPROACH VERSUS NB AND WB-CODED SPEECH SIGNALS

CCR Condition	CMOS	CI_{95}
AMR vs. AMR-WB	2.15	$[2.03; 2.26]$
HMM/GMM vs. AMR-WB	1.48	$[1.35; 1.61]$
DNN-R vs. AMR-WB	1.31	$[1.18; 1.44]$
HMM/GMM vs. **DNN-R**	0.13	$[0.01; 0.24]$
AMR vs. **HMM/GMM**	0.81	$[0.60; 1.03]$
AMR vs. **DNN-R**	1.37	$[1.22; 1.51]$

CMOS and respective 95% confidence interval (CI_{95}) for each of the CCR conditions are presented in Table V.

As expected, the AMR-WB condition outperforms the AMR (NB) condition by a clear 2.15 CMOS points. Relative to the ABE schemes, however, AMR-WB is only 1.48 (**HMM/GMM**) and 1.31 (**DNN-R**) CMOS points better, respectively. Direct comparison between the **HMM/GMM** ABE baseline and the **DNN-R** ABE shows a just significant 0.13 CMOS advantage for the latter. Relative to AMR (NB), the **HMM/GMM** baseline ABE is 0.81 CMOS points better, while the new **DNN-R** ABE improves the coded NB speech by an impressive 1.37 CMOS points, to the authors' knowledge the best reported ABE CCR test results in literature. Being 1.31 CMOS points below AMR-WB, but 1.37 CMOS points above AMR (NB) we can conclude that the **DNN-R** ABE approach bridges about half of the quality gap between coded NB and WB speech.

VII. Conclusion

In this work, we evaluated deep neural network (DNN)-based wideband (WB) spectral envelope estimators in the context of artificial speech bandwidth extension (ABE) and compared the results to a typical baseline HMM/GMM ABE. Several DNN topologies and training strategies have been evaluated, showing that a regression DNN approach with filterbank features and pretrained sigmoid units outperforms the classification-based approaches depending on a codebook and an HMM. For the best DNN-based statistical model cepstral distances could be improved by 1.18 dB, while WB-PESQ indicates a speech quality gain of 0.23 MOS points vs. the HMM/GMM baseline. In a subjective CCR listening test the superiority of using DNNs in ABE was proven by a clear 1.37 CMOS points advantage over AMR-coded narrowband speech.

References

[1] N. R. French and J. C. Steinberg, "Factors governing the intelligibility of speech sounds," *J. Acoust. Soc. Amer.*, vol. 19, no. 1, pp. 90–119, Jan. 1947.

[2] W. Krebber, "Sprachübertragungsqualität von Fernsprech-Handapparaten," (in German), Ph.D. dissertation, Inst. für Nachrichtentechnik, Aachen Univ., Aachen, Germany, (vol. 10, no. 357 of VDI Fortschrittsberichte), 1995.

[3] P. Bauer, M.-A. Jung, J. Qi, and T. Fingscheidt, "On improving speech intelligibility in automotive hands-free systems," in *Proc. IEEE Int. Symp. Consumer Electron.*, Braunschweig, Germany, Jun. 2010, pp. 1–5.

[4] P. Bauer, J. Jones, and T. Fingscheidt, "Impact of hearing impairment on fricative intelligibility for artificially bandwidth-extended telephone speech in noise," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 7039–7043.

[5] J. Abel *et al.*, “A subjective listenining test of six different artificial bandwidth extension approaches in English, Chinese, German, and Korean,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 5915–5919.

[6] J. Makhoul and M. Berouti, “High-frequency regeneration in speech coding systems,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Washington, DC, USA, Apr. 1979, vol. 4, pp. 428–431.

[7] H. Carl and U. Heute, “Bandwidth enhancement of narrow-band speech signals,” in *Proc. Eur. Signal Process. Conf.*, Edinburgh, U.K., Sep. 1994, pp. 1178–1181.

[8] T. Unno and A. McCree, “A robust narrowband to wideband extension system featuring enhanced codebook mapping,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2005, pp. 805–808.

[9] J. Sadasivan, S. Mukherjee, and C. S. Seelamantula, “Joint dictionary training for bandwidth extension of speech signals,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 5925–5929.

[10] A. H. Nour-Eldin and and P. Kabal, “Memory-based approximation of the Gaussian mixture model framework for bandwidth extension of narrowband speech,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Florence, Italy, Aug. 2011, pp. 1185–1188.

[11] Y. Wang, S. Zhao, Y. Yu, and J. Kuang, “Speech bandwidth extension based on GMM and clustering method,” in *Proc. 5th Int. Conf. Commun. Syst. Netw. Technol.*, Apr. 2015, pp. 437–441.

[12] P. Jax and P. Vary, “Wideband extension of telephone speech using a hidden Markov model,” in *Proc. IEEE Workshop Speech Coding*, Delavan, WI, USA, Sep. 2000, pp. 133–135.

[13] P. Bauer and T. Fingscheidt, “A statistical framework for artificial bandwidth extension exploiting speech waveform and phonetic transcription,” in *Proc. Eur. Signal Process. Conf.*, Glasgow, U.K., Aug. 2009, pp. 1839–1843.

[14] I. Katsir, D. Malah, and I. Cohen, “Evaluation of a speech bandwidth extension algorithm based on vocal tract shape estimation,” in *Proc. Int. Workshop Acoust. Signal Enhancement*, Aachen, Germany, Sep. 2012, pp. 1–4.

[15] C. Yagli, M. A. T. Turan, and E. Erzin, “Artificial bandwidth extension of spectral envelope along a Viterbi path,” *Speech Commun.*, vol. 55, pp. 111–118, Jan. 2013.

[16] P. Bauer, J. Abel, and T. Fingscheidt, “HMM-based artificial bandwidth extension supported by neural networks,” in *Proc. Int. Workshop Acoust. Signal Enhancement*, Juan les Pins, France, Sep. 2014, pp. 1–5.

[17] T. Fingscheidt and P. Bauer, “A phonetic reference paradigm for instrumental speech quality assessment of artificial speech bandwidth extension,” in *Proc. 4th Int. Workshop Perceptual Quality Syst.*, Vienna, Austria, Sep. 2013, pp. 36–39.

[18] J. Kontio, L. Laaksonen, and P. Alku, “Neural network-based artificial bandwidth expansion of speech,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 873–881, Mar. 2007.

[19] H. Pulakka and P. Alku, “Bandwidth extension of telephone speech using a neural network and a filter Bank implementation for highband mel spectrum,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2170–2183, Sep. 2011.

[20] G. E. Hinton *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[21] R. Peharz, G. Kapeller, P. Mowlaee, and F. Pernkopf, “Modeling speech with sum-product networks: Application to bandwidth extension,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 2014, pp. 3699–3703.

[22] M. Zöhrer, R. Peharz, and F. Pernkopf, “On representation learning for artificial bandwidth extension,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Sep. 2015, pp. 791–795.

[23] K. Li and C.-H. Lee, “A deep neural network approach to speech bandwidth expansion,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Brisbane, Australia, Apr. 2015, pp. 4395–4399.

[24] B. Liu, J. Tao, Z. Wen, Y. Li, and D. Bukhari, “A novel method of artificial bandwidth extension using deep architectures,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Sep. 2015, pp. 2598–2602.

[25] Y. Gu, Z.-H. Ling, and L.-R. Dai, “Speech bandwidth extension using bottleneck features and deep recurrent neural networks,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, San Francisco, CA, USA, Sep. 2016, pp. 297–301.

[26] Y. Gu and Z. H. Ling, “Restoring high frequency spectral envelopes using neural networks for speech bandwidth extension,” in *Proc. Int. Joint Conf. Neural Netw.*, Killarney, Ireland, Jul. 2015, pp. 1–8.

[27] Y. Wang, S. Zhao, W. Liu, M. Li, and J. Kuang, “Speech bandwidth expansion based on deep neural networks,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Sep. 2015, pp. 2593–2597.

[28] J. Abel, M. Strake, and T. Fingscheidt, “Artificial bandwidth extension using deep neural networks for spectral envelope estimation,” in *Proc. Int. Workshop Acoust. Signal Enhancement*, Xi’an, China, Sep. 2016, pp. 1–5.

[29] J. Makhoul, “Linear prediction: A tutorial review,” *Proc. IEEE*, vol. 63, no. 4, pp. 561–580, Apr. 1975.

[30] B. Fodor and T. Fingscheidt, “Reference-free SNR measurement for narrowband and wideband speech signals in car noise,” in *Proc. 10th ITG Conf. Speech Commun.*, Braunschweig, Germany, Sep. 2012, pp. 199–202.

[31] P. Jax, “Enhancement of bandlimited speech signals: Algorithms and theoretical bounds,” Ph.D. dissertation at RWTH Aachen University, Institut für Nachrichtengeräte und Datenverarbeitung (vol. 15 of P. Vary (ed.), Aachener Beiträge zu digitalen Nachrichtensystemen), 2002.

[32] B. Pfister and T. Kaufmann, *Sprachverarbeitung*. Berlin Germany: Springer, 2008.

[33] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, 1st ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[34] J. Garofolo *et al.*, *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Philadelphia, PA, USA: Linguistic Data Consortium, 1993.

[35] A. Moreno *et al.*, “SpeechDat-Car: A large database for automotive environments,” in *Proc. Int. Conf. Lang. Resources Eval.*, Athens, Greece, May 2000, pp. 1–6.

[36] *Multi-Lingual Speech Database for Telephonometry*, NTT Advanced Technology Corporation, San Jose, CA, USA, 1994.

[37] *Software Tool Library 2009 User’s Manual*, ITU-T Recommendation G.191, Nov. 2009.

[38] *Mandatory Speech Codec Speech Processing Functions: AMR Speech Codec; Transcoding Functions, 3GPP TS 26.090, Rel. 6*, 3GPP; TSG SA, Dec. 2004.

[39] J. E. Markel and A. H. Gray, *Linear Prediction of Speech*. Secaucus, NJ, USA: Springer-Verlag, 1982.

[40] P. Bauer, T. Fingscheidt, and M. Lieb, “Phonetic analysis and redesign perspectives of artificial speech bandwidth extension,” in *Proc. Conf. Electron. Speech Signal Process.*, Frankfurt, Germany, Sep. 2008, pp. 215–223.

[41] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley-Interscience, 2000.

[42] G. Montavon, G. B. Orr, and K.-R. Müller, Eds., *Neural Networks: Tricks of the Trade* (ser. Lecture Notes in Computer Science), 2nd ed. New York, NY, USA: Springer, 2012.

[43] G. E. Hinton, “A practical guide to training restricted Boltzmann machines,” Dept. Comput. Sci., , Univ. Toronto, Toronto, ON, Canada, Tech. Rep. UTML TR 2010–003, 2010.

[44] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, vol. 28, 2013, pp. 1–6.

[45] S. P. Bengio, Y. and P. Frasconi, “Learning longterm dependencies with gradient descent is difficult,” *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[47] R. Hagen, “Spectral quantization of cepstral coefficients,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Adelaide, SA, Australia, Apr. 1994, pp. 509–512.

[48] *Wideband Extension to Recommendation P.862 for the Assessment of Wideband Telephone Networks and Speech Codecs*, ITU-T Recommendation P.862.2, Nov. 2007.

[49] S. Möller *et al.*, “Speech quality prediction for artificial bandwidth extension algorithms,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Lyon, France, Aug. 2013, pp. 3439–3443.

[50] P. Bauer, C. Guillaumé, W. Tirry, and T. Fingscheidt, “On speech quality assessment of artificial bandwidth extension,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 2014, pp. 6082–6086.

[51] J. Abel, M. Kaniewska, C. Guillaumé, W. Tirry, and T. Fingscheidt, “An Instrumental quality measure for artificially bandwidth-extended speech signals,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 2, pp. 384–396, Feb. 2017.

[52] *Methods for Subjective Determination of Transmission Quality*," ITU-T Recommendation P.800, Aug. 1996.
[53] *Speech Codec Speech Processing Functions: AMR Wideband Speech Codec; Transcoding Functions (3GPP TS 26.190, Rel. 6)*, 3GPP; TSG SA, Dec. 2004.
[54] *Transmission Characteristics for Wideband Digital Loudspeaking and Hands-Free Telephony Terminals*, ITU-T Recommendation P.341, Mar. 2011.
[55] *Objective Measurement of Active Speech Level*, ITU-T Recommendation P.56, Dec. 2011.

Johannes Abel received the M.Sc. degree in computer and communications systems engineering from Technische Universität Braunschweig, Braunschweig, Germany. During his studies, he worked as a Student Assistant in the field of speech enhancement and wrote his master thesis at the Institute for Communications Technology, Technische Universität Braunschweig, on artificial bandwidth extension for automatic speech recognition. In 2013, he started working toward the Ph.D. degree in the field of artificial bandwidth extension for telephony applications. His research interests include speech enhancement, machine learning, and automatic speech recognition.

Tim Fingscheidt (S'93–M'98–SM'04) received the Dipl.-Ing. degree in electrical engineering in 1993 and the Ph.D. degree in 1998, both from RWTH Aachen University, Aachen, Germany. He further conducted his research on joint speech and channel coding as a consultant in the Speech Processing Software and Technology Research Department, AT&T Labs, Florham Park, NJ, USA. In 1999, he joined the Signal Processing Department, Siemens AG (COM Mobile Devices), Munich, Germany, and contributed to speech codec standardization in ETSI, 3GPP, and ITU-T. In 2005, he joined Siemens Corporate Technology, Munich, leading the speech technology development activities in recognition, synthesis, and speaker verification. Since 2006, he has been a Full Professor with the Institute for Communications Technology, Technische Universität Braunschweig, Braunschweig, Germany. His research interests include speech and audio signal processing, enhancement, transmission, recognition, and instrumental quality measures. He received several awards, including a prize of the Vodafone Mobile Communications Foundation in 1999 and the 2002 prize of the Information Technology branch of the Association of German Electrical Engineers (VDE ITG), where he is leading the Speech Acoustics Committee ITG FA4.3 since 2015. From 2008 to 2010, he was an Associate Editor for the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING. He has been a member of the IEEE Speech and Language Processing Technical Committee since 2011.