A Generic Neural Acoustic Beamforming Architecture for Robust Multi-Channel Speech Processing

Jahn Heymann[a,*], Lukas Drude[a], Reinhold Haeb-Umbach[a]

[a] *Paderborn University, Warburger Straße 100, Paderborn, Germany*

Abstract

Acoustic beamforming can greatly improve the performance of Automatic Speech Recognition (ASR) and speech enhancement systems when multiple channels are available. We recently proposed a way to support the model-based Generalized Eigenvalue beamforming operation with a powerful neural network for spectral mask estimation. The enhancement system has a number of desirable properties. In particular, neither assumptions need to be made about the nature of the acoustic transfer function (e.g., being anechonic), nor does the array configuration need to be known. While the system has been originally developed to enhance speech in noisy environments, we show in this article that it is also effective in suppressing reverberation, thus leading to a generic trainable multi-channel speech enhancement system for robust speech processing. To support this claim, we consider two distinct datasets: The CHiME 3 challenge, which features challenging real-world noise distortions, and the Reverb challenge, which focuses on distortions caused by reverberation. We evaluate the system both with respect to a speech enhancement and a recognition task. For the first task we propose a new way to cope with the distortions introduced by the Generalized Eigenvalue beamformer by renormalizing the target energy for each frequency bin, and measure its effectiveness in terms of the PESQ score. For the latter we feed the enhanced signal to a strong DNN back-end and achieve state-of-the-art ASR results on both datasets. We further experiment with different network architectures for spectral mask estimation: One small feed-forward network with only one hidden layer, one Convolutional Neural Network and one bi-directional Long Short-Term Memory network, showing that even a small network is capable of delivering significant performance improvements.

Keywords: Robust Speech Recognition, Acoustic Beamforming,
Multi-channel Speech Enhancement, Deep Neural Network

[*] Corresponding author
Email address: jahnheymann@gmail.com (Jahn Heymann)

1. Introduction

Acoustic beamforming has been considered as a front-end processing technique for Automatic Speech Recognition (ASR) for many years. As early as 1990 Compernolle et al. showed that significant word error rate (WER) improvements are achievable by acoustic beamforming [1]. Research on acoustic beamforming has made great progress since then, including the use of novel objective functions, such as the multi-channel Wiener filter, and the consideration of arbitrary Acoustic Transfer Functions (ATFs) from the speech source to the microphones, thus giving up the assumption of an anechoic delay-only propagation path, see, e.g., [2] for a tutorial.

While these modern beamforming concepts have been employed for speech communication tasks, their use as a front-end in ASR was rather limited. Further, with the recent success of ASR back-ends relying on Deep Neural Networks (DNNs), the front-end acoustic beamforming needs reconsideration.

An obvious approach to handle multi-channel signals is to first employ a conventional beamforming approach to condense the multiple signals into one signal which is then fed into a DNN back-end. Delcroix et al. have shown that a strong DNN back-end can be significantly improved with a sophisticated beamformer based on the Minimum Variance Distortionless Response (MVDR) criterion [3].

While this work showed the effectiveness of acoustic beamforming in a DNN-based ASR system, only few multi-channel approaches exist which directly employ DNNs. Swietojanski et al. employed the logarithmic Mel filterbank features of multiple acoustic channels as a parallel input to a Convolutional Neural Network (CNN). They explored different weight sharing approaches and found that channel-wise convolution followed by a cross-channel max-pooling performed better than multi-channel convolution [4]. This approach, however, has the intrinsic drawback that the information on the relative phases between the channels is lost, since current feature extraction methods are agnostic to the phase. On the other hand it is well-known that in geometrically compact microphone array configurations the main difference between the signals of the individual channels reside in their phases, not in their magnitudes.

An alternative approach to make use of multiple input channels for ASR is to leverage temporal difference information between channels by directly working on the raw waveform, i.e., feeding the time domain signals into the DNN. Hoshen et al. reported noticeable performance gains over single-channel input [5]. Following works are even able to achieve better results than a MVDR beamformer [6, 7].

Others proposed to jointly train a MVDR beamformer and the acoustic model [8]. Thereby, they use a DNN to estimate the beamforming weights for the MVDR beamformer given the Time Differences of Arrival (TDOA), perform the beamforming operation, extract the features and finally use these features to train an acoustic model. During this training, they are able to backpropagate the cross-entropy error down to the network estimating the beamforming weights.

In this paper we adhere to the conventional approach of first condensing multiple input channels to a single enhanced output signal to be fed to the ASR back-end. However, we still make use of the recent progress in DNNs by employing a neural network component in the estimation of the beamformer coefficients. We consider the acoustic beamformer to be a multiple-input single-output (MISO) linear time-invariant filter. A key concern is how to estimate the filter coefficients to extract the target signal while suppressing interferences, exploiting the different spatial and spectral properties of the target and the distortions. For the Delay-and-Sum Beamformer (DSB), the filter coefficients can be derived from an estimate of the Direction-of-Arrival (DoA), if the geometry of the microphone array is known. Note that the assumption underlying the DSB is that of an anechoic acoustic environment. If reverberation is to be accounted for, the (relative) ATFs between source and sensors are estimated, which usually requires an estimation of the statistics of the target speech signal [9]. Further, advanced beamforming concepts also require an estimate of the Cross-Power Spectral Density (PSD) matrix of the noise signal.

These statistics can be obtained by estimating spectral masks for speech and noise which are typically obtained by model-based methods, i.e. [10, 11, 12, 13, 14, 15, 16]. Instead of using a model-based approach, we recently proposed to use a DNN to estimate those masks. A distinctive advantage of the proposed neural network based mask estimation is that we explicitly account for time and frequency dependencies during mask estimation whereas most model-based approaches treat individual frequencies independently. This improves the accuracy of the estimated signal statistics and hence the overall results [17]. Additionally, making no assumptions about the distribution of the data for masking but rather inferring it from the training data, we expect this approach to be more robust against different noise types and reverberation. Further, by carrying out mask estimation for each channel separately and relying on microphone array independent signal statistics renders the trained neural network parameters independent of the microphone array configuration. Thus our approach can be applied to arbitrary array configurations. We can even cope with array configurations at test time, which are different from those at training time but still employ a powerful DNN in the multi-channel processing pipeline.

DNNs for mask estimation have been used in single channels speech enhancement for a while (e.g. [18]) and even extended to include the phase [19]. Although similar, the overall concept is different. In speech enhancement, the mask obtained from the DNN is directly applied to filter the signal. Whereas here we still want to calculate the filter using our beamforming model. The DNN becomes a component of this model by specifying the parts of the signal we want to put attention to when calculating the statistics necessary for our beamforming operation.

In [20] and [17] we already considered such a setup with speech distorted by additive noise. Here we employ the very same mask estimation and beamforming concept for speech degraded by reverberation. To this end we assume that the direct signal component and early reflections are the target signal and the remaining signal components are the distortion. This split of the received signal is carried out by convolving the clean speech training data with the early and late part of the ATF, respectively. Note that it is not unrealistic to assume that the acoustic impulse response is available, because for reverberant speech recognition the training data is usually generated by convolving clean speech with either a simulated or a measured impulse response. This is necessary because a training corpus consisting of true recordings in reverberant environments is usually not available.

This article is organized as follows: In Sec. 2 we recapitulate our beamforming concept, which is based on the Generalized Eigenvalue (GEV) beamformer and show different ways to incorporate the estimated masks and to cope with the distortions introduced by the beamformer. Sec. 3 gives a detailed description of the mask estimator networks and their training. Finally, in Sec. 4, we present ASR and speech enhancement results for the CHiME 3 and Reverb database.

2. Acoustic beamforming

In this work, we model an observed signal $\mathbf{Y}$ in the Short Time Fourier Transform (STFT)-domain as the superposition of the target image $\mathbf{X}$ and a distortion $\mathbf{N}$. These distortions might be introduced by noise sources or by reverberation effects:

$$\mathbf{Y}(t, f) = \mathbf{X}(t, f) + \mathbf{N}(t, f), \tag{1}$$

where $t \in \{1, \dots T\}$ is the time frame index and $f \in \{1, \dots F\}$ is the frequency bin index.

To suppress the distortions, we use the GEV beamformer which maximizes the signal-to-noise ratio (SNR) of the beamformer output in each frequency bin separately, leading to the beamformer coefficients [21]:

$$\mathbf{F}_{\text{GEV}}(f) = \underset{\mathbf{F}(f)}{\operatorname{argmax}} \frac{\mathbf{F}(f)^{\text{H}} \mathbf{\Phi}_{\mathbf{XX}}(f) \mathbf{F}(f)}{\mathbf{F}(f)^{\text{H}} \mathbf{\Phi}_{\mathbf{NN}}(f) \mathbf{F}(f)}. \tag{2}$$

$\mathbf{\Phi}_{\mathbf{XX}}(f)$ is the target and $\mathbf{\Phi}_{\mathbf{NN}}(f)$ the noise PSD matrix for the f-th frequency. Please note that this does not require any assumptions regarding the nature of the ATF from the speech source to the sensors or regarding the spatial correlation of the noise [21].

The name of the beamformer stems from the fact that the maximization of the Rayleigh coefficient given in Eq. (2) is achieved by solving a generalized eigenvalue problem: The optimal filter coefficient vector $\mathbf{F}_{\text{GEV}}$ is given by the eigenvector corresponding to the largest eigenvalue of the following generalized eigenvalue problem:

$$\mathbf{\Phi}_{\mathbf{XX}} \mathbf{F} = \lambda \mathbf{\Phi}_{\mathbf{NN}} \mathbf{F}. \tag{3}$$

Hence, the solution only relies on the signal statistics, namely the target PSD matrix $\mathbf{\Phi}_{\mathbf{XX}}$ and the noise PSD matrix $\mathbf{\Phi}_{\mathbf{NN}}$. Note that Eq. (3) does not impose a constraint on the norm of $\mathbf{F}$, and since each frequency is considered independently, this can introduce arbitrary speech distortions.

In the following we consider three approaches to handle this issue. The first is to just ignore the distortions and normalize the principal component for each frequency bin to unit length. This is a valid approach if the final goal is speech recognition. As long as the same normalization is carried out in training and recognition the acoustic model can absorb the speech distortions.

However for speech enhancement applications the distortions are unacceptable. In [21] the following single channel post filter to be applied to the GEV output signal has been derived:

$$g_{\mathrm{BAN}}(f) = \frac{\sqrt{\mathbf{F}_{\mathrm{GEV}}(f)^{\mathrm{H}}\mathbf{\Phi}_{\mathbf{NN}}(f)\mathbf{\Phi}_{\mathbf{NN}}(f)\mathbf{F}_{\mathrm{GEV}}(f)/D}}{\mathbf{F}_{\mathrm{GEV}}(f)^{\mathrm{H}}\mathbf{\Phi}_{\mathbf{NN}}(f)\mathbf{F}_{\mathrm{GEV}}(f)}, \tag{4}$$

where D is the number of microphones. This filter performs a so-called Blind Analytic Normalization (BAN) to obtain a distortionless response in the direction of the speaker: The overall ATF from the target source to the post filter output should have unit gain for every frequency bin. If this were achieved perfectly, speech distortions would be removed and one would eventually arrive at the MVDR beamformer [22, 23].

In this work we propose a third method. We assume to have a reliable estimate of the target mask and can thus estimate the power of the target signal in each frequency. Now, to minimize the distortions introduced by the beamformer, we normalize the beamforming output to match this power distribution over all frequencies.

Why did we not use a MVDR beamformer, which is known to provide a distortionless response in the first place? The reasons are fourfold: First, the GEV concept provides an elegant way to estimate the beamformer coefficients without explicitly estimating the (relative) ATF from the source to the sensors (although this is done implicitly). When using the MVDR the ATF would have to be estimated explicitly. Second, the MVDR requires the computation of the inverse of the noise PSD matrix, while this is avoided in the GEV formulation of Eq. (3). We observed that this matrix inverse can lead to numerical problems, in particular in frequency bins sparsely populated. Third, if the ATF estimate is inprecise, arbitrary distortions are introduced nevertheless. Fourth, although the MVDR has been sucessully used in a similar setting [14], we observed overall worse performance compared to the GEV beamformer [17].

The solution to the GEV problem of Eq. (3) requires the knowledge of the the PSD matrices of the target and the noise signal. These are unknown in general and need to be estimated.

One way to estimate them is to employ non-overlapping masks, $M_{\mathbf{X}}$ for the target signal and $M_{\mathbf{N}}$ for the distortion, respectively, and to calculate the weighted sum of outer products of the microphone signals [24]:

$$\mathbf{\Phi}_{\nu\nu}(f) = \sum_{t=1}^{T} M_{\nu}(t,f)\mathbf{Y}(t,f)\mathbf{Y}(t,f)^{\mathrm{H}}, \tag{5}$$

where $\nu \in \{\mathbf{X}, \mathbf{N}\}$. Here $\mathbf{Y}(t, f)$ is the vector of microphone signals at time frame t and frequency bin f.

Using $M_{\mathbf{N}}$ in Eq. (5) yields $\mathbf{\Phi}_{\mathbf{NN}}$. For the target PSD matrix $\mathbf{\Phi}_{\mathbf{XX}}$ we consider two variants. The first assumes that speech is sparse in the STFT domain and that for the masked time frequency (tf)-bins the target is predominant and the contribution from the distortion can be neglected. Then a tf-bin can be attributed to either speech or noise. Under this assumption and the assumption that speech and noise are uncorrelated, $\mathbf{\Phi}_{\mathbf{XX}}$ can simply be calculated using

$$\mathbf{\Phi}_{\mathbf{XX}} = \sum_{t=1}^{T} M_{\mathbf{X}}(t, f)\mathbf{Y}(t, f)\mathbf{Y}(t, f)^{\mathrm{H}}. \tag{6}$$

However, one may argue that the noise signal is not sparse and that a tf-bin populated by speech may still exhibit a noise component. This leads, again under the assumption that speech and noise are uncorrelated, to the second variant where we calculate the target PSD as

$$\mathbf{\Phi}_{\mathbf{XX}} = \sum_{t=1}^{T} M_{\mathbf{X}}(t, f)\mathbf{Y}(t, f)\mathbf{Y}(t, f)^{\mathrm{H}} - \mathbf{\Phi}_{\mathbf{NN}} \tag{7}$$

3. Neural mask estimator

The previous section showed that we can estimate the necessary statistics for acoustic beamforming by masking the observed signals. As mentioned in the introduction, we obtain these masks with a neural network [1].

In order to be independent of the microphone configuration, we opt to estimate the masks on each input channel separately, however sharing the weight matrices and bias vectors of the neural network for all channels. We thus estimate D speech and D noise masks. Those have to be condensed to a single mask for speech and noise for the estimation of the PSD matrices. This pooling could be done by e.g. averaging, taking the maximum/minimum value, taking the median etc. After some informal, preliminary experiments we chose the median operation. The main reason for this choice is that it is immune to broken channels up to a certain extent. In the used CHiME 3 database for example, some channels did not record anything and the mask estimator classified all tf bins to belong to noise. If we used minimum pooling in those cases the mask for speech would be all zero. The opposite happens on channels where the speech mask is very dense and we then use the maximum. In both cases, using the average would also lead to distorted masks. The median, however, is not affected.

In this work, we investigate three different network types for mask estimation. The simplest one is a feed-forward (FF) network with just one hidden layer. For the second configuration, we extend this network with an additional

[1] All networks in this work have been realized using Chainer [25]. An implementation to train a mask estimator on CHiME 3 is available: https://github.com/fgnt/nn-gev

Table 1: Network configurations for mask estimation

	FF	CNN	BLSTM
L1	5643x513 (ReLU)	32x10x11 (CNN)	513x256 (BLSTM)
L2	513x1026 (Sigmoid)	3168x513 (ReLU)	256x513 (ReLU)
L3	-	513x513 (ReLU)	513x513 (ReLU)
L4	-	513x1026 (Sigmoid)	513x1026 (Sigmoid)

CNN and FF layer at the bottom to increase the modeling capacity while still keeping the amount of parameters reasonable. The third network type uses a recurrent layer, namely a bi-directional Long Short-Term Memory (BLSTM) layer, at the bottom to allow for arbitrary time context length. An overview of the three different architectures is given in Tbl 1.

The input to the FF and the CNN network is a window of 11 frames of the magnitude spectra of one channel. For the STFT we use a frame size of 1024 and a frame shift of 256 (at a sampling rate of 16 kHz). Due to its recurrent nature, the BLSTM network can exploit temporal dependencies of arbitrary length and does not need multiple input frames at the same time.

The last layer of the network has always 1026 units and is split into two parts: The first 513 units estimate the target mask $\text{IBM}_{\mathbf{X}}$, while the last 513 units estimate the noise mask $\text{IBM}_{\mathbf{N}}$. We do not force the values of the estimated masks to be one or zero. Rather, we restrict them to be in the range between one and zero using a Sigmoid non-linearity activation function of both estimates. We also do not enforce non-overlapping masks or masks which sum to one for each time-frequency-bin in any way.

3.1. Weight initialization & optimization

We initialize all layers using a uniform distribution, e.g. $W \sim \mathcal{U}\left[-a, a\right]$. For the BLSTM layer, a is 0.04, while for the Rectified Linear Unit (ReLU) layers and the last layer $a = \sqrt{6}/\sqrt{n_{\text{in}} + n_{\text{out}}}$ [26]. n_{in} is the number of inputs and n_{out} the number of units of the layer. The biases are all initialized with zeros.

We employ ADAM [27] for training. A fixed learning-rate of 0.001 and, for the BLSTM network, full backpropagation through time [28] is used. Additionally, if the norm of a gradient for this network is greater than one, we divide the gradient by its norm [29].

To achieve a better generalization, we use dropout for the input-hidden connection of the BLSTM units [30] and for the input of the ReLU layers [31]. The dropout rate is fixed at $p_{\text{dropout}} = 0.5$ for every layer during the whole training. We never use dropout for the last layer. We use the development data for cross-validation, stopping the training when the loss does not decrease anymore after 5 epochs of patience.

3.2. Normalization

Instead of normalizing the input, we apply the batch normalization [32] for each layer. Here, the activation $\mathbf{a} = \mathbf{W}\mathbf{u} + \mathbf{b}$ is replaced with a normalized activation $\mathrm{BN}\left(\mathbf{a}\right)$, where

$$\mathrm{BN}\left(\mathbf{a}\right) = \gamma\hat{\mathbf{u}} + \beta \tag{8}$$

and

$$\hat{\mathbf{u}}(k) = \frac{\mathbf{W}_k\mathbf{u}(k) - \mathrm{E}\left[\mathbf{W}_k\mathbf{u}(k)\right]}{\sqrt{\mathrm{Var}\left[\mathbf{W}_k\mathbf{u}(k)\right]}}. \tag{9}$$

with γ and β as learnable parameters. The normalization is carried out separately for each unit k.

In contrast to the method proposed in [32], we do not use the population estimates for the mean and variance at decoding time. This is done in the above paper to ensure that the output depends on the input in a deterministic way. Changing the mini-batch size or using differently composed mini-batches would lead to different outputs. In our case however, during test time, one mini-batch (as seen by the layer inputs) comprises the features of one utterance. Hence, we can just use the batch statistics and still get a deterministic output.

3.3. Ideal binary masks as targets

We use ideal binary masks as training targets which are defined as:

$$\mathrm{IBM}_{\mathbf{N}}(t,f) = \begin{cases} 1, & \frac{||\mathbf{X}(t,f)||}{||\mathbf{N}(t,f)||} < 10^{\mathrm{th}_{\mathbf{N}}(f)}, \\ 0, & \text{else,} \end{cases} \tag{10}$$

and

$$\mathrm{IBM}_{\mathbf{X}}(t,f) = \begin{cases} 1, & \frac{||\mathbf{X}(t,f)||}{||\mathbf{N}(t,f)||} > 10^{\mathrm{th}_{\mathbf{X}}(f)}, \\ 0, & \text{else.} \end{cases} \tag{11}$$

respectively.

The two thresholds $\mathrm{th}_{\mathbf{X}}$ and $\mathrm{th}_{\mathbf{N}}$ are not identical. Their values range from -5 to 10 depending on the frequency and are hand-tuned. They are chosen such that a decision in favor of speech/noise is only taken if the instantaneous SNR is high/low enough to ensure a low false acceptance rate. This ensures more reliable cross-power spectral density matrix estimates at the expense of discarding some tf-bins which are categorized to be neither speech nor noise.

If the only source of noise is the reverberant speech (like for the Reverb challenge), $\mathbf{N}$ is not clearly defined. We assume that the direct speech and its early reflections are beneficial, i.e., form the target signal, while the late reverberations are detrimental. When calculating the masks, we consider everything attributed to the first 50 ms of the room impulse response as the target signal and everything else as noise.

3.4. Loss function

The network is trained on all utterances and all channels using the binary cross-entropy cost. With $\mathrm{IBM}_\nu(t,f)$, $\nu \in \{\mathbf{X}, \mathbf{N}\}$ being the target mask and $M_\nu(t,f)$ the networks estimation for that value, the loss is given by

$$\begin{aligned} L = -\frac{1}{F}\frac{1}{2T} \sum_{\nu \in \{\mathbf{N},\mathbf{X}\}} \sum_{t=1}^{T} \sum_{f=1}^{F} \{ & \mathrm{IBM}_\nu(t,f) \log_2 M_\nu(t,f) \\ & + (1 - \mathrm{IBM}_\nu(t,f)) \log_2 (1 - M_\nu(t,f)) \} \end{aligned} \tag{12}$$

To accelerate the training procedure for the BLSTM network, we process each channel in parallel, exploiting the fact that they have the same amount of time steps.

4. Experimental results

4.1. Databases

To evaluate the robustness of the approach described above, we use two distinct datasets. One is from the Reverb challenge and the other one is from the CHiME 3 challenge.

The dataset form the Reverb challenge [33] contains SimData and RealData with a vocabulary size of 5000 words. For the SimData utterances by 28 different speakers are taken from the WSJCAM0 corpus [34] and are convolved with three different room impulse responses. Noise is added at a signal-to-noise ratio of 20 dB. The RealData set consists of 372 utterances from the MC-WSJ-AV corpus [35]. These are a set of WSJCAM0 utterances rerecorded with real speakers in a noisy and reverberant room. The set is divided into a far and a near set with distances of ∼ 100 cm and ∼ 250 cm. The training set consists of simulated data only. This leads to a significant mismatch condition between training and testing with RealData. Not only are the reverberation times different and the audio signal recorded rather than simulated, it is also a different database with different speakers and recording characteristics.

The dataset from the third CHiME challenge [36] features real and simulated 6-channel audio data of prompts taken from the 5k WSJ0-Corpus [37] with 4 different types of real-world background noise. Details are described on the challenge website[2] and in [36].

4.2. Speech enhancement

To assess the speech enhancement performance, we use the Perceptual Evaluation of Speech Quality (PESQ) measure [38]. Since we need the clean target audio data for the evaluation, experiments can only be carried out on the simulated data from the CHiME 3 and Reverb challenge. In the following we

[2] http://spandh.dcs.shef.ac.uk/chime_challenge/data.html

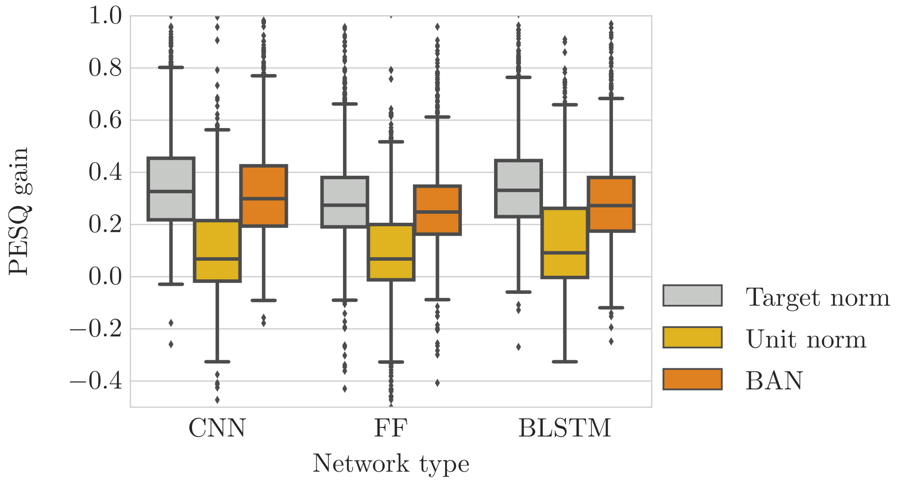

Figure 1: PESQ gain for different networks on the simulated CHiME 3 development data. The average PESQ value for the unprocessed data is 1.27.

compare the different networks for mask estimation and the different normalizations for the beamforming vector **F**.

For all plots, the top and the bottom of the box display the first and third quartile, respectively. The line inside the box shows the median. The whiskers indicate the data which lies within the 1.5 interquartile range. Dots above and below indicate outliers.

Fig. 1 shows the PESQ improvement in comparison to the 5-th input channel for the CHiME 3 data. The differences between the three network types are fairly small, with the CNN delivering the best gains on average. As expected, the variant where the beamforming vector is normalized to unit norm achieves hardly any improvements. Listening to the results, one can observe a high-pass characteristic of the enhanced signal. This is understandable because the distorting noise has a low-pass characteristic. Shifting the signal power from low to higher frequencies is, thus, beneficial in terms of maximizing the output SNR, which is the objective function of the GEV beamformer.

Postprocessing the output can compensate for this effect and lead to better perceptual quality as measured by PESQ. In this noisy scenario, the proposed normalization using the target power distribution over frequencies leads to slightly better results than the BAN. An informal listening test confirms the improvements by BAN and "target norm": The speech sounds more natural as the lower frequencies contain more power after the postprocessing.

The results for a reverberant environment are depicted in Fig. 2. We separate the results according to the three different rooms available in the SimData. The simulated rooms 1, 2, and 3 were considered examples of small, medium and large-size rooms with room reverberation times (T_{60}) of about 0.25 s, 0.5 s, 0.7 s,

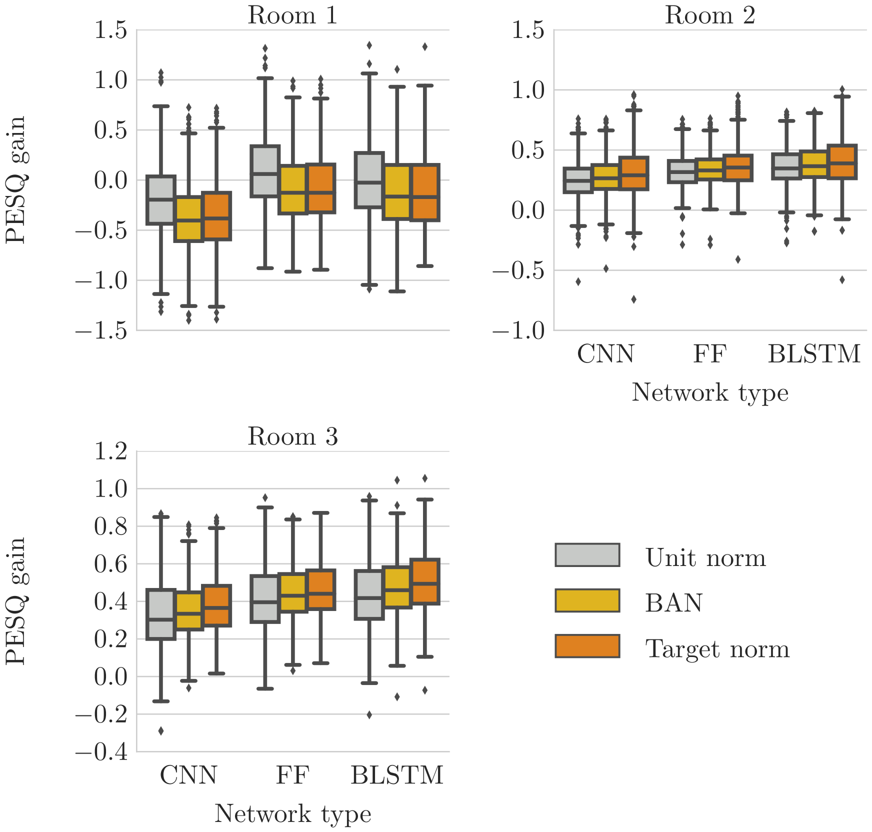

Figure 2: PESQ gain on the simulated Reverb development data. The average PESQ values for the unprocessed data are 2.2 for Room 1, 1.37 for Room 2 and 1.32 for Room 3.

respectively.

For room 1 we can see that the PESQ score output of the beamformer is slightly worse than the observed signal in terms of perceptual quality. This can be attributed to the very short T_{60} time of this room. There is hardly any late reverberation and there are barely any tf-bins dominated by reverberation. For room 2 and room 3 we get a different picture. Here, we get a measurable gain. In contrast to the noisy case, the BLSTM performs best, especially for the room with the biggest reverberation. The CNN is noticeable worse in this condition. Again, the postprocessing brings a performance gain and the ”target norm” normalization achieves the highest score. But the gain compared to the unit norm is far less than in the noisy condition. Also, listening to the results, the unit norm result sounds less distorted. This is understandable because here the distortion, i.e., the late reverberation, does not have the clear low-pass characteristics as the noise in the case of the CHiME 3 data. Additionally, the signal-to-distortion ratio (SDR) for the CHiME 3 utterances is far lower than for the Reverb ones to begin with.

Overall we conclude that a gain is achieved in both, noisy and reverberant conditions, and that the proposed normalization based on the target power distribution over frequency achieves the best results with BAN being only slightly worse.

4.3. Speech recognition

In order to evaluate the speech recognition performance on both datasets we use their respective Kaldi [39] baseline systems. Restricted by the available computational resources, we conduct the comparison of the different networks and normalizations with a computationally less demanding HMM-GMM system. To show the full capability of our approach, we additionally train a strong acoustic model (configuration "VD(X)" from [40]). The network consists of 8 convolutional layers and 3 fully connected layers on top of them. The convolutional layers all have a filter size of 3×3 and the number of channels increases from 3 at the input (40 dimensional log-mel filterbank features + their delta and delta-delta) to 64 and then doubles with every second convolution until it reaches 512 for the last convolutional layer. For further details we refer the reader to [40]. This back-end is used in combination with the supposedly best configuration, the BLSTM mask estimator with unit norm normalization and Eq. (6). For decoding we use three different language models. The first one is the regular WSJ 3-gram model which we also used for the other experiments. The other two, a 5-gram Kneser-Ney [41] and a Recurrent Neural Network (RNN) language model, are used for rescoring and trained on the data provided with the WSJ-database.

For the CHiME 3 data, the HMM-GMM system uses speaker adaptive training, the 3-gram language model from the WSJ-database and BeamformIt [42] as a preprocessing step to use the multiple channels. The Reverb system does not include the speaker adaptive training and the beamforming step. It exploits the fact that corresponding clean data is available and uses the alignment from a model trained on this data as an initialization for the multi-condition training. Other than that, it is comparable to the CHiME 3 system.

Since we are testing many different configurations, we chose to only report the results achieved on the *real* test data for both datasets to keep the tables clear. Nevertheless, parameter-tuning is still done on the development set and the results for the simulated data were within expectations for each configuration (e.g. better than the results on the real data by almost a constant factor.)

4.3.1. Comparison of networks and normalizations

Tbl 2 shows the WER for different configurations for the CHiME 3 data. Again, there is only a minor performance difference regarding the network type used for mask estimation. The BLSTM achieves the best result, but the margin

Table 2: WER on the real evaluation data of the CHiME 3 challenge for different neural networks, target PSD estimations and beamforming normalizations.

network	Eq. (6) + BAN	Eq. (6)	Eq. (7)	Eq. (6) + target norm
BLSTM	16.6	14.8	14.4	17.4
CNN	16.5	14.8	14.9	16.9
FF	16.2	15.2	15.4	16.7
Baseline			23.0	

Table 3: WER on RealData of the Reverb challenge for different neural networks, target PSD estimations and beamforming normalizations.

distance	network	Eq. (6) + BAN	Eq. (6)	Eq. (7)	Eq. (6) + target norm
far	BLSTM	20.5	20.6	20.8	21.9
	CNN	20.7	20.2	21.1	21.5
	FF	20.0	20.0	20.0	19.1
near	BLSTM	18.9	19.3	19.3	20.3
	CNN	19.6	20.0	20.9	19.9
	FF	17.1	19.0	18.0	16.9
far	*Baseline*			42.0	
near				43.5	

is small. Regarding the normalization, the differences are larger. Compared to the speech enhancement results, the performance for the different normalizations are inverted. The unit norm achieves the best results while the target norm is significantly worse. This shows that perceptual speech quality and recognition results are only loosely related and distortions introduced by the front-end do not necessarily harm overall system performance. Concerning the calculation of $\mathbf{\Phi}_{\mathbf{XX}}$, there is hardly a performance difference between the use of Eq. (6) and Eq. (7). It is, however, worth noting, that we can improve upon the baseline by 37% just by exchanging the beamformer.

Like with the speech enhancement, the result for the reverberant environment are slightly different as can be seen in Tbl 3. The difference between the normalization methods vanishes with BAN now achieving best performance, albeit by a very small margin. Interestingly, the simple FF now performs best. Again, we achieve a significant improvement of around 50% compared to the baseline, which, however, does not exploit the multiple channels. But even compared with recently published works who do exploit the multiple channels the error rates are very competitive.

Table 4: WER for the CHiME 3 database with different language models.

		3-gram WSJ	5-gram KN	RNN
simu	dev	7.0	5.8	5.1
	test	7.1	5.8	5.0
real	dev	6.9	5.5	4.6
	test	9.5	7.9	6.6

Table 5: WER for the Reverb database with different language models.

		Room	3-gram WSJ	5-gram KN	RNN
RealData	far		13.7	12.2	10.9
	near		13.5	12.0	10.6
SimData	far	1	4.64	4.15	3.27
		2	5.69	5.30	3.97
		3	6.19	5.27	4.32
	near	1	4.29	4.03	3.15
		2	5.13	4.44	3.54
		3	4.90	4.33	3.51

4.3.2. Results with strong back-end

As mentioned before, the results discussed until this point are generated using a HMM-GMM system. However, we not only want to analyze the behavior of different configurations, but also want to explore the limits of our approach. To this extent, we combine it with a strong CNN-based back-end. The achieved results for the two datasets are reported in Tbl 4 and Tbl 5 respectively. Compared to the HMM-GMM system with the same language model, we achieve a performance gain of 34% for CHiME 3 and around 24% for Reverb. This shows that the beamforming approach does not eat up the improvements achievable further down the processing chain. Also, to the best of our knowledge, these are the best results reported so far for these datasets with utterance-wise decoding (i.e., without adapting the model on the test data). Note, that we only use one beamforming operation and the back-end to achieve these results. No further enhancement is needed and we neither adapt to the speaker nor to the environment. We also did not modify or extend the training data in any way for the mask estimator and the back-end. Regarding the sensitivity to a mismatch condition between training and testing the results show that there is indeed a big difference between the RealData and the SimData. But the results of the challenge[3] show that all systems exhibit this gap. We conclude

[3] http://reverb2014.dereverberation.com/result_asr.html

that the proposed approach is not more sensitive to mismatch conditions than other approaches. However, work remains to be done to close the existing gap.

5. Conclusions

Having shown the effectiveness of our neural network supported beamforming approach for noisy environments before, this article demonstrates that the approach also generalizes well to reverberant environments. The transfer to a distant speech recognition task did not require any modifications of the system architecture. The only difference is in the definition of the distortion, which is now given by the late reverberation. Once the training targets are properly defined, no modification needed to be made to any of the system components, neither the neural network based mask estimation nor the beamformer or the ASR back-end. Not even parameters needed to be adjusted when moving from a noisy to a reverberant ASR task, thus demonstrating impressively the power of learning systems.

The proposed neural network supported beamformer can be used both for speech communication purposes and for ASR. However, we noticed that a configuration which is optimal with respect to perceptual quality is not necessarily optimal for ASR, confirming previous results from other research groups. When the enhancement is combined with a strong ASR back-end very competitive word error rates are achieved both on the CHiME 3 and the Reverb challenge data.

6. Acknowledgements

This work was in part supported by the Deutsche Forschungsgemeinschaft (DFG) under contract no. Ha3455/11-1. Calculations leading to the results presented here were performed on resources provided by the Paderborn Center for Parallel Computing.

References

[1] D. Compernolle, W. Ma, F. Xie, M. Diest, Speech recognition in noisy environments with the aid of microphone arrays 9 (5) (1990) 433 – 442. doi:http://dx.doi.org/10.1016/0167-6393(90)90019-6.
URL http://www.sciencedirect.com/science/article/pii/0167639390900196

[2] S. Gannot, E. Habets, Linear and parametric microphone array processing, https://www.audiolabs-erlangen.de/fau/professor/habets/activities/ICASSP-2013/ (May 2013).

[3] M. Delcroix, T. Yoshioka, A. Ogawa, Y. Kubo, M. Fujimoto, N. Ito, Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the reverb challenge, in: REVERB Challenge Workshop, 2014.

[4] P. Swietojanski, A. Ghoshal, S. Renals, Convolutional neural networks for distant speech recognition 21 (9) (2014) 1120–1124.

[5] Y. Hoshen, R. J. Weiss, K. W. Wilson, Speech acoustic modeling from raw multichannel waveforms, in: Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, 2015, pp. 4624–4628. doi:10.1109/ICASSP.2015.7178847.

[6] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, Andrew, Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms, in: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015, pp. 30–36. doi:10.1109/ASRU.2015.7404770.

[7] T. Sainath, R. Weiss, K. Wilson, A. Narayanan, M. Bacchiani, Factored spatial and spectral multichannel raw waveform cldnns, in: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, 2016.

[8] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. Seltzer, G. Chen, Y. Zhang, M. Mandel, D. Yu, Deep beamforming networks for multi-channel speech recognition, in: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, 2016.

[9] S. Gannot, D. Burshtein, E. Weinstein, Signal enhancement using beamforming and nonstationarity with applications to speech 49 (8) (2001) 1614–1626. doi:10.1109/78.934132.

[10] H. Sawada, S. Araki, S. Makino, Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment 19 (3) (2011) 516–527.

[11] N. Ito, S. Araki, T. Yoshioka, T. Nakatani, Relaxed disjointness based clustering for joint blind source separation and dereverberation, in: Acoustic Signal Enhancement (IWAENC), 2014 14th International Workshop on, 2014, pp. 268–272. doi:10.1109/IWAENC.2014.6954300.

[12] D. H. T. Vu, R. Haeb-Umbach, Blind speech separation employing directional statistics in an expectation maximization framework, in: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, 2010, pp. 241–244. doi:10.1109/ICASSP.2010.5495994.

[13] N. Ito, S. Araki, T. Nakatani, Permutation-free convolutive blind source separation via full-band clustering based on frequency-independent source presence priors, in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 2013, pp. 3238–3242. doi:10.1109/ICASSP.2013.6638256.

[14] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, T. Nakatani, The ntt chime-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices, in: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015, pp. 436–443. doi:10.1109/ASRU.2015.7404828.

[15] S. Araki, T. Nakatani, Hybrid approach for multichannel source separation combining time-frequency mask with multi-channel wiener filter, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 225–228. doi:10.1109/ICASSP.2011.5946381.

[16] S. Araki, M. Okada, T. Higuchi, A. Ogawa, T. Nakatani, Spatial correlation model based observation vector clustering and mvdr beamforming for meeting recognition, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 385–389. doi:10.1109/ICASSP.2016.7471702.

[17] J. Heymann, L. Drude, R. Haeb-Umbach, Neural network based spectral mask estimation for acoustic beamforming, in: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, 2016.

[18] A. Narayanan, D. Wang, Ideal ratio mask estimation using deep neural networks for robust speech recognition, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 7092–7096.

[19] D. S. Williamson, Y. Wang, D. Wang, Complex ratio masking for joint enhancement of magnitude and phase, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 5220–5224.

[20] J. Heymann, L. Drude, A. Chinaev, R. Haeb-Umbach, Blstm supported gev beamformer front-end for the 3rd chime challenge, in: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015, pp. 444–451. `doi:10.1109/ASRU.2015.7404829`.

[21] E. Warsitz, R. Haeb-Umbach, Blind acoustic beamforming based on generalized eigenvalue decomposition 15 (5) (2007) 1529–1539. `doi:10.1109/TASL.2007.898454`.

[22] B. D. Van Veen, K. M. Buckley, Beamforming techniques for spatial filtering.

[23] U. K. Simmer, J. Bitzer, C. Marro, Post-filtering techniques, in: Microphone Arrays, Springer, 2001, pp. 39–60.

[24] M. Souden, S. Araki, K. Kinoshita, T. Nakatani, H. Sawada, A multichannel mmse-based framework for speech source separation and noise reduction, IEEE Transactions on Audio, Speech, and Language Processing 21 (9) (2013) 1913–1928. `doi:10.1109/TASL.2013.2263137`.

[25] S. Tokui, K. Oono, S. Hido, J. Clayton, Chainer: a next-generation open source framework for deep learning, in: Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS), 2015.
URL `http://learningsys.org/papers/LearningSys_2015_paper_33.pdf`

[26] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics, 2010.

[27] D. Kingma, J. Ba, Adam: A method for stochastic optimization.

[28] P. J. Werbos, Backpropagation through time: what it does and how to do it 78 (10) (1990) 1550–1560. `doi:10.1109/5.58337`.

[29] R. Pascanu, T. Mikolov, Y. Bengio, Understanding the exploding gradient problem abs/1211.5063.
URL `http://arxiv.org/abs/1211.5063`

[30] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization abs/1409.2329.
URL `http://arxiv.org/abs/1409.2329`

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting 15 (2014) 1929–1958.
URL `http://jmlr.org/papers/v15/srivastava14a.html`

[32] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift abs/1502.03167.
URL `http://arxiv.org/abs/1502.03167`

[33] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, A. Sehr, W. Kellermann, R. Maas, The reverb challenge: Acommon evaluation framework for dereverberation and recognition of reverberant speech, in: Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on, 2013, pp. 1–4. `doi:10.1109/WASPAA.2013.6701894`.

[34] T. Robinson, J. Fransen, D. Pye, J. Foote, S. Renals, Wsjcamo: a british english speech corpus for large vocabulary continuous speech recognition, in: Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on, Vol. 1, 1995, pp. 81–84 vol.1. `doi:10.1109/ICASSP.1995.479278`.

[35] M. Lincoln, I. McCowan, J. Vepa, H. K. Maganti, The multi-channel wall street journal audio visual corpus (mc-wsj-av): specification and initial experiments, in: Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on, 2005, pp. 357–362. `doi:10.1109/ASRU.2005.1566470`.

[36] J. Barker, R. Marxer, E. Vincent, S. Watanabe, The third chime speech separation and recognition challenge: Dataset, task and baselines, in: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015, pp. 504–511. `doi:10.1109/ASRU.2015.7404837`.

[37] J. Garofalo, et al., CSR-I (WSJ0) complete.

[38] A. W. Rix, J. G. Beerends, M. P. Hollier, A. P. Hekstra, Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs, in: Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on, Vol. 2, 2001, pp. 749–752 vol.2. `doi:10.1109/ICASSP.2001.941023`.

[39] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al., The kaldi speech recognition toolkit, in: IEEE 2011 workshop on automatic speech recognition and understanding, no. EPFL-CONF-192584, IEEE Signal Processing Society, 2011.

[40] T. Sercu, C. Puhrsch, B. Kingsbury, Y. LeCun, Very deep multilingual convolutional neural networks for LVCSR abs/1509.08967.
URL `http://arxiv.org/abs/1509.08967`

[41] R. Kneser, H. Ney, Improved backing-off for m-gram language modeling, in: Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on, Vol. 1, 1995, pp. 181–184 vol.1. `doi:10.1109/ICASSP.1995.479394`.

[42] X. Anguera, C. Wooters, J. Hernando, Acoustic beamforming for speaker diarization of meetings 15 (7) (2007) 2011–2022. doi:10.1109/TASL.2007.902460.