Unsupervised training of neural mask-based beamforming

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

Paderborn University, Department of Communications Engineering, Paderborn, Germany
{drude, heymann, haeb}@nt.upb.de

Abstract

We present an unsupervised training approach for a neural network-based mask estimator in an acoustic beamforming application. The network is trained to maximize a likelihood criterion derived from a spatial mixture model of the observations. It is trained from scratch without requiring any parallel data consisting of degraded input and clean training targets. Thus, training can be carried out on real recordings of noisy speech rather than simulated ones. In contrast to previous work on unsupervised training of neural mask estimators, our approach avoids the need for a possibly pre-trained teacher model entirely. We demonstrate the effectiveness of our approach by speech recognition experiments on two different datasets: one mainly deteriorated by noise (CHiME 4) and one by reverberation (REVERB). The results show that the performance of the proposed system is on par with a supervised system using oracle target masks for training and with a system trained using a model-based teacher.

Index Terms: deep learning, multi-channel, unsupervised learning, robust speech recognition, beamforming

1. Introduction

Despite great progress in acoustic modeling and other fields of automatic speech recognition (ASR), multi-channel front-end processing remains an important factor to achieve high recognition rates in far-field scenarios, such as those encountered by digital home assistants with a spoken language interface. Beamforming is the classic approach to multi-channel ASR. It is used to steer a beam of increased sensitivity towards a desired speech source, thus suppressing interferers with different spatial characteristics. In the currently predominant approaches for blind beamforming, the beamformer coefficients are obtained by estimating the spatial covariance matrices of the desired source signal and the interferences [1, 2, 3].

To obtain these matrices, the sparsity of speech in the short time Fourier transform (STFT) domain is exploited, by which each time frequency (tf)-bin can be described by containing either speech and noise or noise only. Traditionally, this classification is accomplished by using either hand-crafted signal presense probability (SPP) estimators or by employing probabilistic spatial mixture models. A particularly impressive system has been used in the CHiME 3 winning contribution, where a time-variant complex Gaussian mixture model (TV-cGMM) is used to inform a beamforming algorithm [4].

However, it turns out that neural networks can be fairly well trained to distinguish between speech and noise tf-bins and can, therefore, yield a discriminatively trained SPP estimator. This led to the development of neural network-based beamforming [1, 2] and can be considered state of the art on the CHiME 4 data now [3]. These estimators are faster during inference, avoid the local (frequency) and global permutation problem, are easier to adapt to a low-latency setting, and have shown to outperform probabilistic mixture models. Although this development has let to more robust systems and has been evaluated also on industry scale datasets [5, 6], it has one important drawback: it relies on parallel data for supervised training. This means that each training utterance must be available in both a clean and a degraded version, the first serving as training target and the latter as network input. This is practically only possible if the distortion is artificially added to the clean recording. As a consequence, certain effects which are hard to simulate, e.g., the Lombard effect [7], are not captured during training. Further, recording clean data and realistic spatial noise is way more expensive than collecting abundant real-world noisy data.

One possibility to train a neural mask estimator without parallel clean data is to train it end-to-end with an ASR criterion, e.g. connectionist temporal classification (CTC) and/or sequence-to-sequence [8], or cross-entropy [9]. But these systems are hard to train [6, 9], do not always reach the performance of their separately trained counterparts [6] and require transcribed data which is again expensive to obtain for real environments. Another option is to generate intermediate masks with an unsupervised teacher, as proposed in e.g. [10, 11], and also in [12] where we demonstrate how to leverage a probabilistic spatial mixture model, namely a complex angular central Gaussian mixture model (cACGMM), to generate intermediate masks. However, this approaches require a – possibly hand-crafted – teacher system and also a lot of computational resources to either store the intermediate masks or generate them on-the-fly.

In contrast, we here directly use a neural mask estimator to initialize the expectation maximization (EM) algorithm of a cACGMM as part of the training. We calculate the likelihood of the multi-channel observations under this model and update the parameters of the neural mask estimator by backpropagating the gradient of the likelihood through the EM algorithm. The main advantage of this is that the spatial model is now part of the processing graph and always gets the best initialization given the most recent network parameters. We show that a single EM iteration per training step is enough, whereas the model in [12] used 100 EM iterations to create the teacher masks.

It is worth noting, that backpropagation into a spatial mixture model has already been demonstrated to work in a supervised setup in [13], where the supervision stems from a first-pass decoding of an acoustic model. Hershey et al. introduced a whole class of new architectures by proposing to backpropagate through any iterative algorithm and to discriminatively update model parameters in each iteration step [14]. This naturally included deep unfolding of non-negative matrix factorization (NMF) and also deep unfolding of complex Gaussian mixture models (cGMMs) [15]. In contrast to [15] we here optimize a mask estimation network which is not part of the EM algorithm. Further, they proposed to train the parameters with a supervision signal as of [15, Eq. 19], whereas we constrain ourselves to unsupervised likelihood training.

[⋆] Both authors contributed equally.

2. Signal model

A D channel recording is modeled in the STFT domain by a D-dimensional vector $\mathbf{y}_{tf}$ at time frame index t and frequency bin index f. In a far-field scenario, this signal is impaired by (convolutive) reverberation and additive noise:

$$\mathbf{y}_{tf} = \mathbf{x}_{tf} + \mathbf{n}_{tf}, \tag{1}$$

where $\mathbf{x}_{tf}$ is the STFTs of the source signal which is convolved with the room impulse response (RIR). The noise term $\mathbf{n}_{tf}$ captures directed and undirected background noise sources.

3. Neural mask-based beamforming

The generalized eigenvalue (GEV) (or Max-SNR) beamformer criterion maximizes the expected output signal to noise ratio (SNR) of the beamforming operation [16]:

$$\mathbf{w}_f^{(\mathrm{GEV})} = \underset{\mathbf{w}_f}{\mathrm{argmax}} \frac{\mathbb{E}\left\{\left|\mathbf{w}_f^{\mathsf{H}}\mathbf{x}_{tf}\right|^2\right\}}{\mathbb{E}\left\{\left|\mathbf{w}_f^{\mathsf{H}}\mathbf{n}_{tf}\right|^2\right\}}. \tag{2}$$

The ratio is maximized by the eigenvector corresponding to the largest eigenvalue of the generalized eigenvalue problem

$$\mathbf{\Phi}_f^{(\mathbf{xx})}\mathbf{w}_f = \lambda\mathbf{\Phi}_f^{(\mathbf{nn})}\mathbf{w}_f, \tag{3}$$

where $\mathbf{\Phi}_f^{(\mathbf{xx})} = \mathbb{E}\left\{\mathbf{x}_{tf}\mathbf{x}_{tf}^{\mathsf{H}}\right\}$ and $\mathbf{\Phi}_f^{(\mathbf{nn})} = \mathbb{E}\left\{\mathbf{n}_{tf}\mathbf{n}_{tf}^{\mathsf{H}}\right\}$ are the spatial covariance matrices of speech and noise, respectively. The solution to this problem is computed for each frequency bin separately. It is unique up to a multiplication with a complex scalar and, thus, arbitrary distortions can be introduced. We compute the solution by decomposing $\mathbf{\Phi}_f^{(\mathbf{nn})}$ with a Cholesky decomposition, resulting in a similar regular eigenvalue problem with a Hermitian matrix. To arrive at the solution of the generalized eigenvalue problem, the resulting eigenvector is projected back with $\mathbf{L}_f^{-\mathsf{H}}$ where $\mathbf{L}_f\mathbf{L}_f^{\mathsf{H}} = \mathbf{\Phi}_f^{(\mathbf{nn})}$. The eigenvector itself is scaled to unit norm such that the scaling is only determined by the noise covariance matrix. To avoid distortions due to the scale of the noise covariance matrix and limit the confusion of the acoustic model back-end which was trained on unprocessed training data we scale the noise covariance matrix as follows:

$$\widetilde{\mathbf{\Phi}}_f^{(\mathbf{nn})} = \mathbf{\Phi}_f^{(\mathbf{nn})} \Big/ \operatorname{tr}\left\{\mathbf{\Phi}_f^{(\mathbf{nn})}\right\}. \tag{4}$$

The beamforming algorithm requires the frequency-dependent covariance matrices of speech and noise, respectively:

$$\mathbf{\Phi}_f^{(kk)} = \sum_t M_{tf}^{(k)}\mathbf{y}_{tf}\mathbf{y}_{tf}^{\mathsf{H}} \Big/ \sum_t M_{tf}^{(k)}, \tag{5}$$

where the masks $M_{tf}^{(k)}$ are estimated from the observed signal using a neural network with a mask indicating for each tf-bin if the speech ($k = \mathbf{x}$) or the noise is predominant ($k = \mathbf{n}$).

The architecture of the mask estimator is the same as the bidirectional long short term memory network (BLSTM) mask estimator in [1] and it also operates on each microphone channel independently. The masks are pooled with a mean operation resulting in a single mask for speech as well as noise as an input to the EM algorithm during training resulting and with a median operation to be used in Eq. 5 during test time. To avoid a transformation back to the time domain prior to feature extraction of the subsequent ASR, the mask estimator as well as the beamformer operate in the spectral domain with an FFT size of 160 (10 ms) a frame size of 400 (25 ms) and a frame shift of 160 (10 ms) specifically tailored to the ASR back-end.

4. Probabilistic spatial mixture models

Based on the assumption that speech is a sufficiently sparse signal in the STFT domain [17, 18] one can model the observations with a mixture model with K classes (here $K = 2$). In its generic form, the distribution of the multi-channel observations can be formulated as a marginalization over all classes with the assumption that all observations are conditionally i.i.d.:

$$p(\mathbf{y}_{tf}) = \sum_k \pi_{kf}\, p(\mathbf{y}_{tf}|\boldsymbol{\theta}_k), \tag{6}$$

where π_{kf} is the a-priori probability, that an observation belongs to mixture component k, and $p(\mathbf{y}_{tf}|\boldsymbol{\theta}_k)$ is any appropriate class conditional distribution which can model $\mathbf{y}_{tf}$, while $\boldsymbol{\theta}_k$ captures all class-dependent parameters.

The cACGMM [19] uses a complex Angular central Gaussian distribution [20] as a class conditional distribution:

$$p(\tilde{\mathbf{y}}_{tf}|\mathbf{B}_{kf}) = \frac{(D-1)!}{2\pi^D \det \mathbf{B}_{kf}} \frac{1}{(\tilde{\mathbf{y}}_{tf}^{\mathsf{H}}\mathbf{B}_{kf}^{-1}\tilde{\mathbf{y}}_{tf})^D}, \tag{7}$$

where $\tilde{\mathbf{y}}_{tf} = \mathbf{y}_{tf}/\|\mathbf{y}_{tf}\|$. Due to this normalization, the model can only capture intra-channel level differences but does not account for the power of an observation. Additionally, it is worth noting, that $\tilde{\mathbf{y}}_{tf}^{\mathsf{H}}\mathbf{B}_{kf}^{-1}\tilde{\mathbf{y}}_{tf}$ is invariant to the absolute phase, thus $p(\tilde{\mathbf{y}}_{tf}) = p(\tilde{\mathbf{y}}_{tf}e^{\mathrm{j}\phi})$. Therefore, the model only captures intra-channel phase differences, but not the absolute phase.

This spatial mixture model neglects frequency dependencies. Thus, when used without any kind of guidance, it will yield a solution where the speaker index is inconsistent over frequency bins. This issue is the so called frequency permutation problem [21]. It can be addressed by calculating that permutation alignment (PA) (bin by bin) which maximizes the correlation of the masks along neighboring frequencies [21][1].

5. Complex backpropagation

To motivate a gradient descent algorithm on a computational graph which involves complex values, we first need to clarify the differentiability of complex-valued functions. A complex function $g : \mathbb{C} \to \mathbb{C}$ is differentiable if the following limit converges to a single value independent of the path of h:

$$\frac{\mathrm{d}g}{\mathrm{d}z} = \lim_{h\to 0} \frac{g(z+h) - g(z)}{h} \tag{8}$$

However, only a certain class of functions is complex differentiable – these functions are called holomorphic. In contrast many relevant building blocks for neural networks, e.g., the cost function, can by definition not be holomorphic (due to its real-only output). An elegant way around this is to make use of Wirtinger calculus, where [22] nicely proved that non-holomorphic functions are still partially differentiable, e.g. the partial differential with respect to the complex conjugate of a complex value can be defined as follows:

$$\frac{\partial g}{\partial z^*} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + \mathrm{j}\frac{\partial f}{\partial y}\right), \tag{9}$$

where $g(z(x,y), z^*(x,y)) = f(x,y)$.

The technical report [23] lists a large number of useful building blocks for complex-valued backpropagation. A large amount of complex-valued operations and their derivatives are now available within TensorFlow [24].

[1] The particular permutation alignment solver we used in this work can be found here: `https://github.com/fgnt/pb_bss`

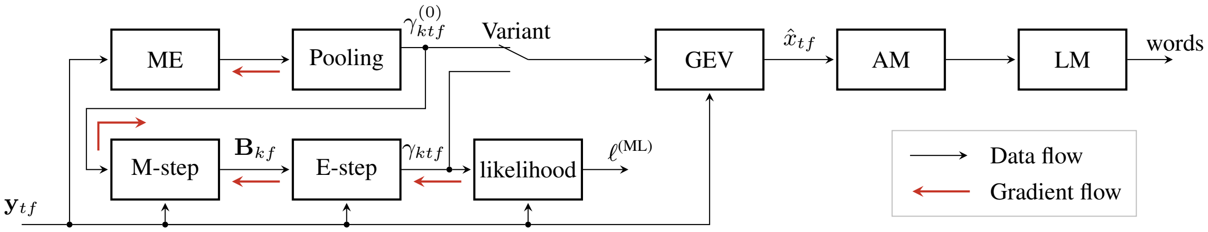

Figure 1: *Overview of the proposed system. The parameters of the mask estimator (ME) are optimized by backpropagating gradients from the likelihood function through the EM algorithm and through the pooling operation into the mask estimator. The beamforming operation (GEV) and the ASR back-end (AM, LM) are not part of the optimization. The training does not require any supervision.*

6. Proposed system

A neural mask estimator is used to calculate intermediate class affiliations $\gamma_{ktf}^{(0)}$ for the two classes speech and noise or noise only. Due to the random initialization of the network parameters, these masks are just as random in the beginning of the training procedure. During training, this one is used to obtain the class-dependent mixture weights π_{kf} and the class-dependent covariance matrices $\mathbf{B}_{kf}$ corresponding to the M-step of the EM algorithm. Consequently, the current values of π_{kf} and $\mathbf{B}_{kf}$ depend on the current utterance and on the network parameters:

$$\pi_{kf} = \frac{1}{T}\sum_t \gamma_{ktf}^{(0)}, \tag{10}$$

$$\mathbf{B}_{kf} = D\sum_t \gamma_{ktf}^{(0)} \frac{\tilde{\mathbf{y}}_{tf}\tilde{\mathbf{y}}_{tf}^{\mathsf{H}}}{\tilde{\mathbf{y}}_{tf}^{\mathsf{H}}\mathbf{B}_{kf}^{-1}\tilde{\mathbf{y}}_{tf}} \Bigg/ \sum_t \gamma_{ktf}^{(0)}. \tag{11}$$

Please note, that Eq. 11 is an implicit definition of $\mathbf{B}_{kf}$ which can be solved again by iterations [19]. However, we here opt to initialize the matrix with an identity matrix and apply Eq. 11 only once per frequency bin. It is worth noting, that we here suggest to use a single M-step followed by a single E-step instead of several iterations of the EM algorithm.

Now, we calculate the observation likelihood [25, Eq. 9.28] under the assumption of this particular spatial mixture model:

$$\ell^{(\mathrm{ML})} = \sum_{t,f} \ln \sum_k \pi_{kf} p(\tilde{\mathbf{y}}_{tf}|\mathbf{B}_{kf}). \tag{12}$$

We now learn the real-valued parameters of the underlying neural mask estimator by backpropagating the gradients of the real-valued likelihood through the complex-valued update equations of the EM algorithm.

For completeness, we also compare the following variants:

$$\ell^{(\mathrm{ML,equal})} = \sum_{t,f} \ln \sum_k \frac{1}{K} p(\tilde{\mathbf{y}}_{tf}|\mathbf{B}_{kf}), \tag{13}$$

$$\ell^{(\mathrm{ML,auxiliary})} = \sum_{k,t,f} \tilde{\gamma}_{ktf} \ln\left(\pi_{kf} p(\tilde{\mathbf{y}}_{tf}|\mathbf{B}_{kf})\right), \tag{14}$$

where $\tilde{\gamma}_{ktf}$ is either the output of the neural network $\gamma_{ktf}^{(0)}$ or the updated affiliations as a result of the E-step γ_{ktf}. Eq. 13 is the likelihood of the observation under the assumption of equal mixture weights. Eq. 14 is the auxiliary function [25, Eq. 9.30].

During training, this training procedure can introduce a frequency permutation problem as described in Sec. 4. Especially in the beginning of the training, this can lead to conflicting gradients when, e.g., the noise class is represented by the first model output in the first example of a batch and the second output in the second example of a batch for a given frequency. This can be alleviated by using a permutation alignment algorithm [21] as explained in Sec. 4. The resulting alignment map can then be used to permute the class affiliations, or – possibly more elegant – to permute the weights of the last layer of the neural mask estimator, directly.

Once the system is trained, the intermediate class affiliations $\gamma_{ktf}^{(0)}$ created by the neural mask estimator can either be used directly for the covariance matrix estimation in Eq. 5 or can be refined by an additional M-step and a subsequent E-step. Once the covariance matrices are obtained, a beamforming vector can be calculated using Eq. 2 which is then used to obtain the speech estimate: $\hat{x}_{tf} = \mathbf{w}_f^{\mathsf{H}}\mathbf{y}_{tf}$.

7. Relation to variational autoencoders

It is worth noting, that this training scheme is fairly reminiscent of how a variational autoencoder (VAE) [26] is trained. Therefore, this section highlights these similiarities.

The loss function of a VAE consists of a negative log-likelihood which describes how well the observation fits to the model and a Kullback-Leibler divergence which measures how well a latent posterior distribution fits to prior assumptions of the latent code. In our context this results in:

$$\begin{aligned} J^{(\mathrm{VAE})} = &- \mathbb{E}_{q(\mathcal{Z}|\mathbf{y}_{tf})}\left\{\ln p\left(\tilde{\mathbf{y}}_{tf}|\mathcal{Z}\right)\right\} \\ &+ \mathrm{KL}\left(q(\mathcal{Z}|\mathbf{y}_{tf})\|p(\mathcal{Z})\right), \end{aligned} \tag{15}$$

where $\mathcal{Z}$ contains the random variables $\gamma_{ktf}^{(0)}$ and $\mathbf{B}_{kf}$ which both depend on the network output. The network now produces the parameters of a Dirichlet distribution which then models the posterior of $\gamma_{ktf}^{(0)}$, while $\mathbf{B}_{kf}$ deterministically depends on $\gamma_{ktf}^{(0)}$ as in Eq. 11. The observation distribution $p(\tilde{\mathbf{y}}_{tf}|\mathcal{Z})$ can now be used in the likelihood term. The estimated values are obtained by sampling from the Dirichlet distribution using the reparameterization trick [26]. The prior in the latent space is assumed to be an uninformative Dirichlet distribution on $\gamma_{ktf}^{(0)}$. This allows to learn an uncertainty estimate of the mask and opens up interesting research questions, e.g., how to better estimate the covariance matrices for beamforming.

8. Acoustic model

Our hybrid acoustic model (AM) is a wide bi-directional residual network (WBRN) as proposed in [27]. It consists of a combination of a wide residual network to model local context and a BLSTM to model long term dependencies. The hyperparameters were adapted from [27]. The choice fell to a WBRN since it is considered state of the art on the single-channel track with baseline RNNLM rescoring during the CHiME 4 challenge. Without rescoring, it reaches a word error rate (WER) of 16.05 % on the real test set. The most recent Kaldi recipe yields 16.34 % [3] without rescoring.

Table 1: *WERs on the real test set of the CHiME 4 challenge database for different loss functions for the unsupervised mask estimator training. The additional EM step determines, if a single EM step is used at inference time.*

Loss function		Additional EM step	
Type	Variant	no	yes
ML, Eq. 12	$\gamma_{ktf}^{(0)}$	8.83	8.25
	γ_{ktf}	8.53	8.05
equal, Eq. 13	$1/K$	8.12	7.80
auxiliary, Eq. 14	$\gamma_{ktf}^{(0)}$	8.68	8.08
	γ_{ktf}	8.82	8.15

9. Evaluation

To assess the performance of the algorithm, we evaluate on two distinct databases, one mainly impaired by noise and one mainly affected by reverberation, both with a sampling rate of 16 kHz. All systems are evaluated with an FFT size of 512 (32 ms), a window size of 400 (25 ms) and a shift of 160 (10 ms).

We first evaluate different variants of the loss function according to Eq. 12 – Eq. 14 by training the mask estimator on the simulated CHiME 4 [28] training set and evaluating on real recordings of the corresponding evaluation set and summarize the results in Tbl. 1. The dataset contains six-channel recordings with microphones mounted on an of the shelf tablet obtained in public spaces. First of all, it becomes apparent that an additional EM step improves the performance over directly using the network output mask for beamforming in all cases. Whether using the neural network output $\gamma_{ktf}^{(0)}$ directly in the loss or using the result of the E-step γ_{ktf} in the loss depends on the particular loss function. The best results are obtained, when assuming equal mixture weights for the underlying probabilistic spatial model.

Next, we compare different training strategies in Tbl. 2. The cACGMM yields a fairly high variance in output quality mainly caused by permutation alignment issues and overall yields a WER of 13.06 % with potential to be tuned further to the particular test set. When the mask estimator is trained with oracle masks as training targets, we the best WER with a sigmoid output nonlinearity and no additional EM step. Using a softmax nonlinearity degrades the WER slightly. When the aforementioned cACGMM is used as a teacher to train the mask estimator as in [12] we obtain almost the same WERs as in the supervised setting with a softmax nonlinearity (7.95 % WER). The proposed system with likelihood training yields a WER of 7.80 % which is close to the supervised performance with a softmax nonlinearity and an additional EM step.

Table 2: *Comparison of training strategies with supervised (with oracle masks) and unsupervised systems (unsupervised teacher, proposed likelihood training) on CHiME 4 real data. Supervised systems are typeset in gray.*

		No add. EM step		Add. EM step	
Estimator	Training	Sigmoid	Softmax	Sigmoid	Softmax
None					16.05
cACGMM					13.06
Neural	Oracle	7.46	7.97	7.75	7.71
	Teacher	7.79	7.95	7.86	7.86
	Likelihood		8.12		7.80

Table 3: *Word error rates on the REVERB challenge real test dataset. Supervised systems are typeset in gray.*

			WPE	
Estimator	Training	Activation	no	yes
None			15.52	11.97
cACGMM			12.66	7.16
Neural	Oracle	Sigmoid	7.86	7.02
		Softmax	7.85	6.87
	Teacher	Softmax	8.37	7.95
	Likelihood	Softmax	7.97	7.96

Tbl. 3 summarizes results obtained on the REVERB [29] database. It contains eight-channel recordings in a reverberant enclosure. This evaluation is particularly interesting, because the main cause of signal degradation is here reverberation. Therefore, we evaluate all algorithms with and without an additional weighted prediction error (WPE) dereverberation algorithm [30, 31, 32] preceeding the entire processing pipeline. First of all, it can be observed that the cACGMM results improve dramatically, when preceded by an intial dereverberation. This seems plausible, since the cACGMM model was derived without particularly modeling convolution in the STFT domain. The best supervised WER is obtained with a softmax activation and WPE. When training the mask estimator with a cACGMM as teacher, the additional dereverberation step still improves the performance. Interestingly, the proposed unsupervised mask estimator yields almost the same WER with and without additional WPE and therefore yields competitive WERs at lower computational costs during training as well as during inference.

10. Conclusions

We presented an unsupervised training scheme for neural mask estimators for acoustic beamforming and therefore eliminate the need for simulated recordings. In contrast to a teacher-student scheme, it does neither require a costly full-fledged EM algorithm, nor excessive disk space to store teacher masks and therefore scales well to large amounts of real recordings. We demonstrated, that the unsupervised training performance is comparable to supervised training. Overall, the proposed unsupervised training scheme is a more streamlined approach, is less computational demanding than other unsupervised approaches, and can be trained entirely on real recordings. This has applications way beyond beamforming and future research will look into multi-speaker scenarios and single-channel speech enhancement.

11. Reproducability instructions

Since there is no fine-tuning of oracle masks necessary, the proposed approach can be reproduced fairly easily using, e.g., TensorFlow. To be able to reproduce the Numpy results of the probabilistic spatial models including models not analyzed here an implementation including permutation alignment can be found at `https://github.com/fgnt/pb_bss`. The implementation of the WPE dereverberation can be found at `https://github.com/fgnt/nara_wpe`.

12. Acknowledgment

Computational resources were provided by the Paderborn Center for Parallel Computing.

13. References

[1] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 196–200.

[2] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 1981–1985.

[3] S.-J. Chen, A. S. Subramanian, H. Xu, and S. Watanabe, "Building state-of-the-art distant speech recognition using the CHiME-4 challenge with a setup of speech enhancement baseline," *arXiv preprint arXiv:1803.10109*, 2018.

[4] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi *et al.*, "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 436–443.

[5] C. Boeddeker, H. Erdogan, T. Yoshioka, and R. Haeb-Umbach, "Exploring practical aspects of neural mask-based beamforming for far-field speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6697–6701.

[6] J. Heymann, M. Bacchiani, and T. N. Sainath, "Performance of mask based statistical beamforming in a smart home scenario," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6722–6726.

[7] M. Garnier, N. Henrich, and D. Dubois, "Influence of sound immersion and communicative interaction on the lombard effect," *Journal of Speech, Language, and Hearing Research*, vol. 53, no. 3, pp. 588–608, 2010.

[8] T. Ochiai, S. Watanabe, T. Hori, J. R. Hershey, and X. Xiao, "Unified architecture for multichannel end-to-end speech recognition with neural beamforming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1274–1288, Dec 2017.

[9] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, "BEAMNET: End-to-end training of a beamformer-supported multi-channel ASR system," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[10] P. Seetharaman, G. Wichern, J. Le Roux, and B. Pardo, "Bootstrapping single-channel source separation via unsupervised spatial clustering on stereo mixtures," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[11] E. Tzinis, S. Venkataramani, and P. Smaragdis, "Unsupervised deep clustering for source separation: Direct learning from mixtures using spatial information," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[12] L. Drude, D. Hasenklever, and R. Haeb-Umbach, "Unsupervised training of a deep clustering model for multichannel blind source separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[13] T. Higuchi, T. Yoshioka, and T. Nakatani, "Optimization of speech enhancement front-end with speech recognition-level criterion," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 3808–3812.

[14] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[15] S. Wisdom, J. Hershey, J. Le Roux, and S. Watanabe, "Deep unfolding for multichannel source separation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 121–125.

[16] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1529–1539, 2007.

[17] M. Aoki, M. Okamoto, S. Aoki, H. Matsui, T. Sakurai, and Y. Kaneda, "Sound source segregation based on estimating incident angle of each frequency component of input signals acquired by multiple microphones," *Acoustical Science and Technology*, vol. 22, no. 2, pp. 149–157, 2001.

[18] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[19] N. Ito, S. Araki, and T. Nakatani, "Complex angular central Gaussian mixture model for directional statistics in mask-based microphone array signal processing," in *European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1153–1157.

[20] J. T. Kent, "Data analysis for shapes and images," *Journal of statistical planning and inference*, vol. 57, no. 2, pp. 181–193, 1997.

[21] H. Sawada, S. Araki, and S. Makino, "Measuring dependence of bin-wise separated signals for permutation alignment in frequency-domain BSS," in *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2007, pp. 3247–3250.

[22] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," in *IEE Proceedings F (Communications, Radar and Signal Processing)*, vol. 130. IET, 1983, pp. 11–16.

[23] C. Böddeker, P. Hanebrink, L. Drude, J. Heymann, and R. Haeb-Umbach, "On the computation of complex-valued gradients with application to statistically optimum beamforming," *arXiv preprint arXiv:1701.00392*, 2017.

[24] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "TensorFlow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[25] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[26] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[27] J. Heymann, L. Drude, and R. Haeb-Umbach, "Wide residual BLSTM network with discriminative speaker adaptation for robust speech recognition," in *International Workshop on Speech Processing in Everyday Environments (CHiME16)*, 2016, pp. 12–17.

[28] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, 2016.

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

[30] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Blind speech dereverberation with multi-channel linear prediction based on short time fourier transform representation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2008, pp. 85–88.

[31] ——, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[32] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A Python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *13. ITG Fachtagung Sprachkommunikation (ITG 2018)*, Oct 2018.