DEEP COMPLEX-VALUED NEURAL BEAMFORMERS

Lukas Pfeifenberger[1]*, Matthias Zöhrer*[1]*, Franz Pernkopf*

Signal Processing and Speech Communication Lab
Graz University of Technology

ABSTRACT

We propose a complex-valued deep neural network (cDNN) for speech enhancement and source separation. While existing *end-to-end* systems use complex-valued gradients to pass the training error to a real-valued DNN used for gain mask estimation, we use the full potential of complex-valued LSTMs, MLPs and activation functions to estimate complex-valued beamforming weights directly from complex-valued microphone array data. By doing so, our cDNN is able to locate and track different moving sources by exploiting the phase information in the data. In our experiments, we use a typical living room environment, mixtures of the WallStreet Journal corpus, and YouTube noise. We compare our cDNN against the BeamformIt toolkit as a baseline, and a mask-based beamformer as a state-of-the-art reference system. We observed a significant improvement in terms of PESQ, STOI and WER.

Index Terms— beamforming, complex-valued deep neural networks, Wirtinger Calculus

1. INTRODUCTION

Recent contributions to data-driven beamforming propose a DNN to estimate a spectral gain mask from noisy, multi-microphone speech signals. This mask is used to obtain the power spectral density (PSD) matrices of the desired and interfering sound sources. With those PSD estimates, statistical beamformers such as the *Minimum Variance Distortionless Response* (MVDR) beamformer [1] or the *Generalized Eigenvalue* (GEV) beamformer [2] are used to estimate the desired signal. DNN-based gain mask estimators have been proposed in [3, 4, 5]. As those approaches use magnitude spectrograms as features, they do not exploit the spatial information contained in the phase of the data. In [6, 7], we circumvent this limitation by using the eigenvectors of the short-time PSD matrix of the noisy speech as features. This allows for a significantly smaller DNN to estimate the gain mask, with comparable performance in both ASR results and perceptual speech quality [7]. However, mask-based beamforming requires an entire block of audio data at a time. During this period, the signal statistics are assumed to be constant. This limits the capability to track moving sound sources. An attempt towards online processing has been proposed in [8], where the PSD matrices are recursively estimated.

With recent trends towards *end-to-end* ASR systems, the DNN-based mask estimator, the beamformer and the acoustic front-end of the ASR system are combined into a fully interconnected model. This allows to back-propagate the training error from the acoustic modelling cost function through the beamformer and the DNN-based mask estimator [9, 10, 11, 12]. As beamforming involves non-holomorphic functions (i.e. conjugation or absolute value), their gradients do not exist. A widely adopted solution for this problem is to split complex-valued functions into their *real* and *imaginary* parts, and treat them like real-valued functions. However, this results in losing important properties like complex rotation or symmetry. Using *Wirtinger Calculus*, it is possible to derive complex-valued gradients from non-holomorphic functions with respect to a real-valued variable [13, 14, 15].

While end-to-end systems make use of the complex-valued gradient of statistical beamformers, they still use a real-valued DNN to estimate the gain mask. We aim to explore the full potential of complex-valued gradients and propose a fully complex DNN (cDNN) beamformer, with complex LSTM and MLP layers, as well as complex-valued activation functions. By doing so, we do not need to rely on a gain mask, as the cDNN is able to predict complex-valued beamforming weights directly from complex-valued microphone signals. Unlike a statistical beamformer, such a model estimates a set of optimal beamforming weights for each time-frequency bin. This leverages the source tracking and separation performance. To demonstrate the capabilities of our cDNN, we perform simulations involving moving and static sound sources in a typical living room setup, using mixtures of the WallStreet Journal corpus (WSJ) [16] and YouTube noise [17]. We compare the performance of the cDNN against a baseline using *BeamformIt* [18] and a reference system using a mask-based beamformer [6] with online tracking [8]. We further report performance metrics, i.e. ΔSNR, PESQ [19], STOI [20], as well as WER scores. Compared to the mask-based beamformer, the proposed system reaches an average relative improvement of 58.47% WER.

[1] Both authors contributed equally.

This work was supported by the Austrian Science Fund (FWF) under the project number I2706-N31 and NVIDIA for providing GPUs.

2. COMPLEX-VALUED MULTI LAYER PERCEPTRONS

A complex-valued MLP (cMLP) is defined analogously to its real-valued counterpart. i.e.

$$\mathbf{h}^{(t)} = g(\mathbf{W}_h \mathbf{z}^{(t)} + \mathbf{b}_h), \tag{1}$$

where $\mathbf{z}^{(t)}$ denotes the input, and $\mathbf{W}_h$ and $\mathbf{b}_h$ are the internal weights and biases, respectively. All variables are defined over $\mathbb{C}$. Based on recent contributions on complex-valued neural networks [21, 22, 23], we propose the non-linear complex-valued activation function $g(\cdot)$ as natural extension of a real-valued *tanh* unit, i.e.

$$g(\mathbf{z}) = \tanh(|\mathbf{z}|) \odot \frac{\mathbf{z}}{|\mathbf{z}|}, \tag{2}$$

where $\odot$ denotes element-wise multiplication. The function $g(\mathbf{z})$ is symmetric, with a magnitude bounded by 1.0. The phase of $\mathbf{z}$ is not modified. For comparison, we demonstrate the behavior of a *tanh* activation function with non-complex gradients (i.e. the real and imaginary parts are stacked and treated as individual values). It is given as $g_2(\mathbf{z}) = \tanh(\mathrm{Re}\{\mathbf{z}\}) + i \tanh(\mathrm{Im}\{\mathbf{z}\})$. Figure 1 shows the magnitude and phase response of $g(\mathbf{z})$ in panel (a) and (b), and the magnitude and phase response of $g_2(\mathbf{z})$ in panel (c) and (d), respectively. It can be seen that $g_2(\mathbf{z})$ is not bounded to 1.0. It also modifies the phase to a constant value per quadrant of the complex plane.

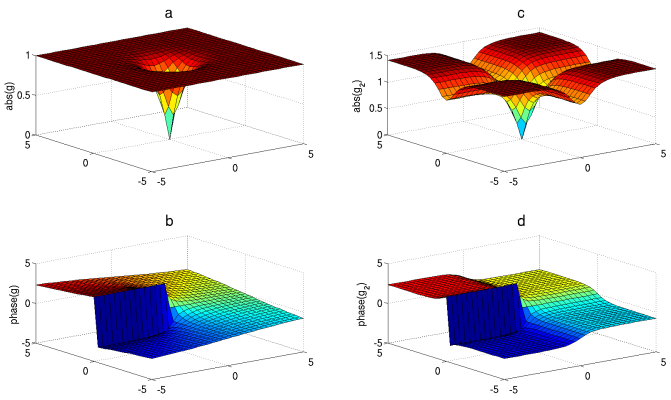

Fig. 1: Magnitude and phase of $g(\mathbf{z})$ and $g_2(\mathbf{z})$.

3. COMPLEX-VALUED LONG SHORT TERM MEMORY NETWORKS

In complex-valued LSTMs (cLSTM) the input $\mathbf{i}^{(t)}$, forget $\mathbf{f}^{(t)}$ and output $\mathbf{o}^{(t)}$ gate are calculated as follows:

$$\mathbf{i}^{(t)} = \sigma\Big(\mathrm{Re}\Big\{\mathbf{W}_{zi}\mathbf{z}^{(t)} + \mathbf{W}_{hi}\mathbf{h}^{(t-1)} + \mathbf{b}_i\Big\}\Big), \tag{3a}$$

$$\mathbf{f}^{(t)} = \sigma\Big(\mathrm{Re}\Big\{\mathbf{W}_{zf}\mathbf{z}^{(t)} + \mathbf{W}_{hf}\mathbf{h}^{(t-1)} + \mathbf{b}_f\Big\}\Big), \tag{3b}$$

$$\mathbf{o}^{(t)} = \sigma\Big(\mathrm{Re}\Big\{\mathbf{W}_{zo}\mathbf{z}^{(t)} + \mathbf{W}_{ho}\mathbf{h}^{(t-1)} + \mathbf{b}_o\Big\}\Big). \tag{3c}$$

Similar to real-valued LSTMs [24], the memory cell $\mathbf{c}^{(t)}$ is updated according to

$$\tilde{\mathbf{c}}^{(t)} = g(\mathbf{W}_{zc}\mathbf{z}^{(t)} + \mathbf{W}_{hc}\mathbf{h}^{(t-1)} + \mathbf{b}_c), \quad \text{and} \tag{4a}$$

$$\mathbf{c}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \tilde{\mathbf{c}}^{(t)}. \tag{4b}$$

The hidden state is determined as

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot g(\mathbf{c}^{(t)}). \tag{5}$$

Figure 2 shows the network graph of the resulting cLSTM. Again, all variables are defined over $\mathbb{C}$. Note that $g_2(\mathbf{z})$ cannot be used in Eq. (5), as its magnitude is greater than one. This would cause the gradient of $\mathbf{h}^{(t)}$ to grow exponentially when using back-propagation through time. The activations $\sigma(\cdot)$ for the gating variables in Eq. (3a) - (3c) are real-valued sigmoid functions, to ensure that the gating mechanism is not altering the phase information of the input signal.

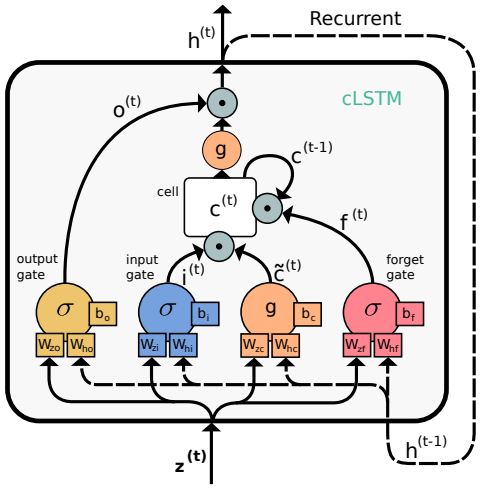

Fig. 2: Complex LSTM unit with internal connections.

We use *Wirtinger Calculus* [13, 25, 15] to obtain complex-valued gradients for each component. It allows us to iteratively apply the chain-rule to complex derivatives, i.e. complex-valued back-propagation. It can also be applied to stochastic gradient descent optimization algorithms like ADAM [26]. For further details on complex-valued back-propagation we refer the interested reader to [11].

4. DEEP COMPLEX-VALUED NEURAL BEAMFORMING

The cDNN uses the complex-valued microphone samples $Z(k,t,m)$ as features, with $k = 1,\ldots,K$ frequency bins and $m = 1,\ldots,M$ microphones. To speed up the learning process, the features are decorrelated using *Principal Component Analysis* (PCA) whitening. For each time frame t, the cDNN processes a matrix of $M \times K$ features $\boldsymbol{Z}(t)$, and predicts a $M \times K$ matrix of complex-valued beamforming weights $\boldsymbol{W}(t)$. The cDNN composed of three cLSTM layers and three cMLP layers with $2MK$ neurons between each hidden layer. The beamforming step is a *filter-and-sum* operation, i.e. $Y(k,t) = \boldsymbol{W}(k,t)^H \boldsymbol{Z}(k,t)$. Figure 3 provides a system overview.

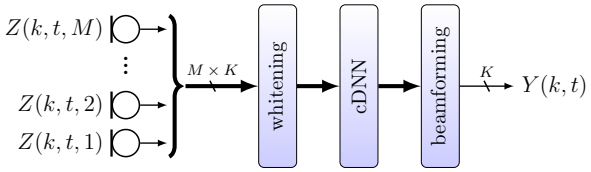

Fig. 3: System overview.

The signal arriving at the microphones is composed of an additive mixture of N sound sources, i.e.

$$\boldsymbol{Z}(k,t) = \sum_{n=1}^{N} \boldsymbol{S}_n(k,t), \tag{6}$$

where $\boldsymbol{S}_n(k,t)$ represents the n^{th} sound source at frequency bin k and time frame t. Each sound source is composed of a monaural recording $X_n(k,t)$ convolved with an *Acoustic Transfer Function* (ATF) denoted by $\boldsymbol{A}_n(k,t)$, i.e.

$$\boldsymbol{S}_n(k,t) = \boldsymbol{A}_n(k,t) X_n(k,t). \tag{7}$$

The ATFs model the acoustic path from a sound source to the microphones, including all reverberations and reflections caused by the room acoustics [27]. To simulate the ATFs for point sources, we use the *Image Source Method* (ISM) [28]. The living room is modeled as shoebox with a reflection coefficient of $\beta = 0.85$ for each wall, and a reflection order of 10. This results in a reverberation time of approximately $250ms$. For static sources, software libraries such as [29] are readily available. For dynamic sources, we generate a new set of ATFs every $32ms$. We also generate an isotropic background noise using

$$\boldsymbol{S}_n(k,t) = \boldsymbol{U}(k,t) X_n(k,t), \tag{8}$$

with $\boldsymbol{U}(k,t) = \boldsymbol{E}(k)\boldsymbol{\Lambda}(k)^{0.5}\, \mathrm{e}^{i\boldsymbol{\varphi}(k,t)}$. The matrices $\boldsymbol{\Lambda}(k)$ and $\boldsymbol{E}(k)$ are the eigenvalues and eigenvectors of the spatial coherence matrix $\boldsymbol{\Gamma}(k)$ for a spherical sound field [30]. The $M \times 1$ vector $\boldsymbol{\varphi}(k,t)$ denotes a uniformly distributed phase between $-\pi,\ldots,\pi$. It can easily be seen that the PSD matrix of $\boldsymbol{S}_n(k,t)$ has the properties of a spherical sound field, i.e. $\mathbb{E}\{\boldsymbol{S}_n(k)\boldsymbol{S}_n^H(k)\} = \boldsymbol{\Gamma}(k)\Phi_{X_nX_n}(k)$, where $\Phi_{XX}(k)$ is the power spectrum of the monaural recording $X_n(k,t)$.

5. EXPERIMENTAL SETUP

To test the performance of our cDNN, we simulate a typical living room scenario with two static speakers S_1 and S_2, a TV set S_3, and two moving speakers D_1 and D_2. The dynamic paths D_1 and D_2 change randomly within a region of 2m on each side, as indicated in Figure 4. To simulate head movements of the static sources S_1 and S_2, random position changes occur within a cube of $20cm$ in size. We use a circular microphone array with $M = 6$ microphones and a diameter of $86mm$, located next to the TV set. Within this environment, we define the five experiments given in Table 1.

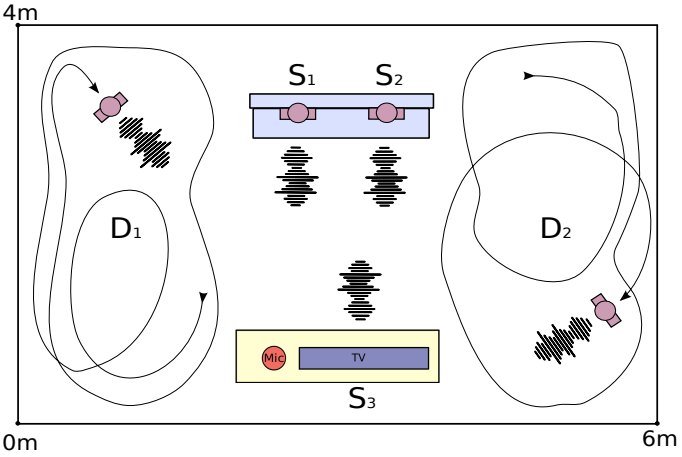

Fig. 4: Shoebox model of a living room showing stationary sound sources S_1 to S_3, and dynamic sound sources D_1 and D_2. The microphone array is located next to the TV set.

Experiment #	Desired source	Interfering source(s)
1	D_1	D_2
2	D_1	isotropic
3	S_1	isotropic
4	S_1	S_3
5	S_2	$\mathsf{D}_1, \mathsf{S}_3$

Table 1: Experimental setups.

For each experiment, the cDNN predicts beamforming weights $\boldsymbol{W}(k,t)$ which preserve the desired source $\boldsymbol{S}_1(k,t)$, and cancel out the interfering sources $\boldsymbol{S}_{2\ldots N}(k,t) = \sum_{n=2}^{N} \boldsymbol{S}_n(k,t)$. The cost function $\mathcal{L}(k,t)$ of the cDNN is designed to maximize the ΔSNR after applying the beamforming weights, i.e.

$$\mathcal{L}(k,t) = 10log_{10}\frac{|\boldsymbol{W}^H\boldsymbol{S}_1|^2}{|\boldsymbol{W}^H\boldsymbol{S}_{2\ldots N}|^2} - 10log_{10}\frac{||\boldsymbol{S}_1||_2^2}{||\boldsymbol{S}_{2\ldots N}||_2^2} \tag{9}$$

for each time-frequency bin[1]. The mean over all time steps T and frequency bins K is then used for back-propagation. Note that the weights $\boldsymbol{W}(k,t)$ do not represent a statistical beamformer like the MVDR or GEV, but rather an optimal filter-and-sum beamformer for each time-frequency bin in a max-SNR fashion. To avoid unbounded weights, we normalize each predicted beamforming vector to unit length, i.e. $|\boldsymbol{W}(k,t)| \stackrel{!}{=} 1$. As a consequence, speech distortions will occur. However, it is possible to control those distortions using *Blind Statistical Normalization* (BSN) [2].

5.1. Training and Testing

For training, we use 12776 utterances from the *si_tr_s* set of the WSJ0 [16] corpus for the speech sources in Eq. (7), and 27 hours of 32 different sound categories from YouTube [17] as isotropic background noise in Eq. (8). All recordings are sampled at 16kHz, and converted to frequency domain with $K = 513$ bins and 50% overlapping blocks. The sources in Eq. (6) are mixed with equal volume. For testing, we use 651 utterances from the *si_et_05* set of the WSJ0 corpus mixed with another 5 hours of Youtube noise of the same 32 categories. For each of the five experiments in Table 1, a separate cDNN and mask-based beamformer is trained.

5.2. Results

We use the *BeamformIt* toolkit as baseline, and the mask-based beamformer in [6] with online tracking from [8] as reference system. For each method and each experiment, we report the ΔSNR from Eq. (9), the *Perceptual Evaluation of Speech Quality* score (PESQ) [19], the *Short-Time Objective Intelligibility* measure (STOI), and the WER obtained by the Google Speech-to-Text API [31]. In particular, the WER was computed using the clean WSJ0 test set as reference, for which the Google Speech-to-Text API reports a WER of 6.1%. From Table 2 it can be seen that BeamformIt performs poorly for experiments with more than one source, i.e. experiments 1, 4 and 5. This is to be expected, as BeamformIt relies on blind DOA estimation to localize a single source. The mask-based beamformer with online tracking shows better performance for those experiments, which has also been observed in [8]. However, our cDNN outperforms this approach significantly, as we are able to estimate the optimal beamformer weights for each time-frequency bin in a max-SNR sense. Figure 5 shows an utterance from the test set using the 1^{st} experiment, where two dynamic sound sources D_1 and D_2 are constantly moving around the living room. Panel (a) shows the mixture $Z(k,t,1)$ for the first microphone, and panel (b) shows the estimate $Y(k,t)$. It can be seen that the cDNN predicts beamforming weights according to the occurrence of the sound sources, i.e. source signal D_1 is preserved, and D_2 is canceled.

[1]For enhanced readability, the indices k, t have been omitted in Eq. (9).

Method	Experiment #	ΔSNR	PESQ	STOI	WER
BeamformIt	1	-	1.325	0.699	76.7%
	2	-	1.222	0.774	17.7%
	3	-	1.222	0.764	17.9%
	4	-	1.179	0.632	43.2%
	5	-	1.186	0.588	88.3%
mask-based BF + online tracking	1	4.445	1.514	0.834	46.1%
	2	4.286	1.576	0.837	32.8%
	3	4.516	1.751	0.866	18.5%
	4	8.690	1.439	0.811	45.6%
	5	7.011	1.402	0.792	58.3%
cDNN	1	6.156	1.688	0.825	21.5%
	2	8.736	2.263	0.882	9.0%
	3	9.558	2.551	0.902	6.1%
	4	10.306	1.652	0.792	13.4%
	5	9.212	1.441	0.758	33.7%

Table 2: Results

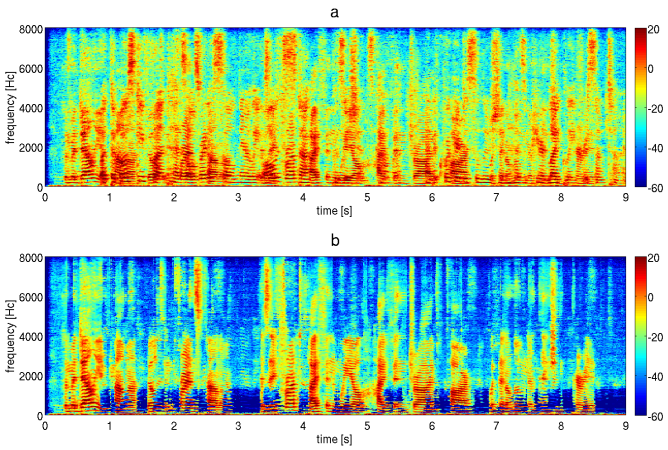

Fig. 5: (a) mixture of two dynamic sound sources D_1 and D_2. (b) separated source D_1 predicted by the cDNN.

6. CONCLUSIONS AND FUTURE WORK

We presented a complex-valued deep neural network (cDNN) to estimate complex-valued beamforming weights directly from complex-valued microphone data. Unlike existing approaches, our cDNN uses fully complex-valued LSTM and MLP layers, as well as complex-valued activation functions. Comparisons against BeamformIt and a state-of-the art mask-based beamforming system showed a significant improvement in terms of ΔSNR, PESQ, STOI and WER. Future work includes experiments on real multi-channel recordings, and the inclusion of our model in an end-to-end system.

7. REFERENCES

[1] B. D. V. Veen and K. M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, no. 5, pp. 4–24, Apr. 1988.

[2] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamform-

ing based on generalized eigenvalue decomposition," *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, Jul. 2007.

[3] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *2016 IEEE ICASSP*, 2016.

[4] J. Heymann, L. Drude, A. Chinaev, and 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)*, Dec 2015, pp. 444–451.

[5] H. Erdogan, J. Hershey, S. Watanabe, M. Mandel, and J. L. Roux, "Improved mvdr beamforming using single-channel mask prediction networks," in *Interspeech*, Sep. 2016.

[6] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, "Dnn-based speech mask estimation for eigenvector beamforming," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, 2017, pp. 66–70.

[7] ——, "Eigenvector-based speech mask estimation using logistic regression," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2017, pp. 2660–2664.

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

[9] Z. Meng, S. Watanabe, J. R. Hershey, and H. Erdogan, "Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 271–275.

[10] T. Menne, R. Schlüter, and H. Ney, "Speaker Adapted Beamforming for Multi-Channel Automatic Speech Recognition," *ArXiv e-prints*, 2018.

[11] 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," *CoRR*, vol. abs/1701.00392, 2017.

[12] 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 5325–5329.

[13] M. F. Amin, M. I. Amin, A. Y. H. Al-Nuaimi, and K. Murase, "Wirtinger calculus based gradient descent and levenberg-marquardt learning algorithms in complex-valued neural networks," in *Neural Information Processing*. Springer Berlin Heidelberg, 2011, pp. 550–559.

[14] P. Bouboulis, "Wirtinger's calculus in general hilbert spaces," *CoRR*, vol. abs/1005.5170, 2010. [Online]. Available: http://arxiv.org/abs/1005.5170

[15] R. F. H. Fischer, "Appendix A: Wirtinger calculus," in *Precoding and Signal Shaping for Digital Transmission*. Wiley-Blackwell, 2005, pp. 405–413.

[16] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '91. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 357–362.

[17] "PyTube – a lightweight, pythonic, dependency-free, library for downloading youtube videos." 2018. [Online]. Available: https://python-pytube.readthedocs.io/en/latest/

[18] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2021, September 2007.

[19] "ITU-T recommendation P.862. Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," 2000.

[20] H.-Y. Dong and C.-M. Lee, "Speech intelligibility improvement in noisy reverberant environments based on speech enhancement and inverse filtering," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2018, no. 1, p. 3, May 2018.

[21] C. Trabelsi, O. Bilaniuk, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," *CoRR*, vol. abs/1705.09792, 2017.

[22] C.-A. Popa, "Complex-valued stacked denoising autoencoders," in *Advances in Neural Networks – ISNN 2018*, T. Huang, J. Lv, C. Sun, and A. V. Tuzikov, Eds. Cham: Springer International Publishing, 2018, pp. 64–71.

[23] M. Tygert, J. Bruna, S. Chintala, Y. LeCun, S. Piantino, and A. Szlam, "A mathematical motivation for complex-valued convolutional networks," *Neural Computation*, vol. 28, no. 5, pp. 815–825, 2016.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] P. Bouboulis and S. Theodoridis, "Extension of wirtinger's calculus to reproducing kernel hilbert spaces and the complex kernel lms," *Trans. Sig. Proc.*, vol. 59, no. 3, pp. 964–978, Mar. 2011.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization." *CoRR*, vol. abs/1412.6980, 2014.

[27] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Berlin–Heidelberg–New York: Springer, 2008.

[28] E. A. P. Habets and S. Gannot, "Generating sensor signals in isotropic noise fields," *The Journal of the Acoustical Society of America*, vol. 122, no. 6, pp. 3464–3470, 2007.

[29] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: A python package for audio room simulations and array processing algorithms," *CoRR*, vol. abs/1710.04196, 2017.

[30] H. Kuttruff, *Room Acoustics*, 5th ed. London–New York: Spoon Press, 2009.

[31] "SpeechRecognition – a library for performing speech recognition, with support for several engines and apis, online and offline." 2018. [Online]. Available: https://pypi.org/project/SpeechRecognition/