FRAME-BY-FRAME CLOSED-FORM UPDATE FOR MASK-BASED ADAPTIVE MVDR BEAMFORMING

Takuya Higuchi, Keisuke Kinoshita, Nobutaka Ito, Shigeki Karita, Tomohiro Nakatani

NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan

ABSTRACT

Beamforming approaches using time-frequency masks have recently been investigated and have shown promising results for noise robust automatic speech recognition (ASR) in many tasks. The time-frequency masks are estimated to compute the spatial statistics of target speech and noise signals, and then the statistics are used to derive a beamformer. Although its effectiveness has been clearly shown in batch and block-wise processing, it has not been well extended to frame-by-frame processing, which is a very important procedure for many actual applications. In this paper, we derive a frame-by-frame update rule for a mask-based minimum variance distortion-less response (MVDR) beamformer, which enables us to obtain enhanced signals without a long delay by combining it with uni-directional recurrent neural network-based mask estimation. Based on the Woodbury matrix identity, our algorithm achieves a closed-form solution of the mask-based MVDR beamformer at every time frame without any matrix inversion. Experimental results show that our frame-by-frame beamformer outperforms baseline block-wise beamforming on the CHiME-3 simulation dataset even with a shorter time delay.

Index Terms— Speech enhancement, beamforming, online processing, time-frequency masking

1. INTRODUCTION

Beamforming is a key technique for speech enhancement and noise robust automatic speech recognition (ASR). In a beamforming scenario, beamformer coefficients are multiplied by multichannel observed signals at each frequency to obtain enhanced signals. For effective beamforming, it is important to estimate appropriate beamformer coefficients.

One approach that can be used to obtain the beamformer coefficients is to parameterize neural networks with the coefficients as in [1]. The coefficients are regarded as some of parameters of the neural networks, i.e., an acoustic model of an ASR system, and are trained on multichannel training data. Another approach is to estimate the coefficients with the neural networks [2, 3], where the neural networks estimate the real and imaginary parts of the coefficients from multichannel input features. However, these approaches cannot be applied to multichannel signals recorded with a different microphone array since the number of neural network parameters depends on the number of microphones.

On the other hand, a mask-based beamforming approach achieves a microphone array independent system by exploiting time-frequency masks [4–8]. First, the single channel time-frequency masks are estimated to allow us to compute spatial statistics with the multichannel observed signals. Then, the statistics can be used to obtain the beamformer coefficients. Several approaches can be employed to obtain the coefficients from the statistics including the max-SNR beamformer [4], and the minimum variance distortion-less response (MVDR) beamformer [5].

The effectiveness of the mask-based beamformer for speech enhancement and noise robust ASR has been described in many articles, however, previous investigations employed batch processing and block-wise processing. For practical applications, it is very important to obtain enhanced signals with a short time delay, and so it is worth investigating frame-by-frame processing with the mask-based beamformer.

In this paper, we derive a frame-by-frame update rule for mask-based MVDR beamformer coefficients, where the time-frequency masks for statistic computation are also estimated frame-by-frame with long short-term memories (LSTMs). Our proposed MVDR beamformer enables us to obtain the enhanced signal frame-by-frame without a long delay. A closed-form solution of the MVDR beamformer can be obtained at every time frame with our algorithm based on the Woodbury matrix identity. This closed-form solution allows us to obtain beamformer coefficients that achieve the global minimum of an objective function of the MVDR beamformer at every time frame without any matrix inversion. Experimental results show that our frame-by-frame beamformer yields a large ASR performance gain from unprocessed signals and outperforms a baseline block-wise beamformer on the CHiME-3 simulated evaluation set even with a shorter time delay.

2. MASK-BASED MVDR BEAMFORMING

This section describes MVDR beamforming with batch processing, where the spatial statistics are estimated based on time-frequency masks. The mask-based MVDR beamformer

978-1-5386-4658-8/18/$31.00 ©2018 IEEE

will be extended for frame-by-frame processing in the next section.

Let us assume that a noisy speech signal is recorded with M microphones. $\mathbf{y}_{f,t} = [y_{1,f,t}, ..., y_{M,f,t}]^{\mathrm{T}}$ denotes an $M \times 1$ dimensional observation at (f, t), where f and t denote the frequency and time indices, respectively. An enhanced speech signal $\hat{s}_{f,t}$ can be obtained by beamforming as

$$\hat{s}_{f,t} = \mathbf{w}_f^{\mathrm{H}} \mathbf{y}_{f,t}, \tag{1}$$

where $\mathbf{w}_f$ denotes an $M \times 1$ dimensional beamformer coefficient at frequency f.

The MVDR beamformer can be derived by minimizing the total power of the beamformer outputs

$$\begin{aligned} \frac{1}{T}\sum_t |\hat{s}_{f,t}|_2^2 &= \frac{1}{T}\sum_t \mathbf{w}_f^{\mathrm{H}} \mathbf{y}_{f,t} \mathbf{y}_{f,t}^{\mathrm{H}} \mathbf{w}_f \\ &= \mathbf{w}_f^{\mathrm{H}} \mathbf{Y}_f \mathbf{w}_f, \end{aligned} \tag{2}$$

with a constraint $\mathbf{w}_f^{\mathrm{H}} \mathbf{h}_f = h_{m_{ref},f}$. $\mathbf{Y}_f = \sum_t \mathbf{y}_{f,t} \mathbf{y}_{f,t}^{\mathrm{H}}/T$ denotes the covariance matrix of the observed signals, and $\mathbf{h}_f = [h_{1,f}, ..., h_{M,f}]^{\mathrm{T}}$ denotes the steering vector of the target signal. $h_{m_{ref},f}$ denotes the transfer gain between the reference microphone and the target source. This linear constraint keeps the beamformer coefficients distortionless in the direction parameterized by $\mathbf{h}_f$. A closed-form solution to minimize the objective function in Eq. (2) with the constraint can be derived as [9]

$$\mathbf{w}_f = \frac{\mathbf{Y}_f^{-1} \mathbf{h}_f}{\mathbf{h}_f^{\mathrm{H}} \mathbf{Y}_f^{-1} \mathbf{h}_f} h_{m_{ref},f}^*. \tag{3}$$

Instead of using the steering vector $\mathbf{h}_f$, we can parameterize the MVDR beamformer coefficients by using the covariance matrix of the target signal as [10]

$$\mathbf{w}_f = \frac{\mathbf{Y}_f^{-1} \mathbf{R}_f^{(s)} \mathbf{d}}{\mathrm{tr}(\mathbf{Y}_f^{-1} \mathbf{R}_f^{(s)})}, \tag{4}$$

where $\mathbf{d}$ denotes a one-hot vector whose m_{ref}-th component is one and the other components are zero. $\mathbf{R}_f^{(s)}$ denotes the covariance matrix of the target signal. Eqs. (3) and (4) are equivalent when $\mathbf{R}_f^{(s)} = \mathbf{h}_f \mathbf{h}_f^{\mathrm{H}}$. We use Eq. (4) for the MVDR beamformer to avoid steering vector extraction with eigenvector decomposition as in [5, 7, 11].

The covariance matrix of the target signal is often unknown, therefore we estimate the covariance matrix by using time-frequency masks as in [4–8]. When we assume the sparsity of the target signal and interference, the covariance matrix of the target speech signal $\mathbf{R}_f^{(s)}$ can be obtained by

$$\mathbf{R}_f^{(s)} = \frac{1}{\sum_t M_{f,t}^{(s)}} \sum_t M_{f,t}^{(s)} \mathbf{y}_{f,t} \mathbf{y}_{f,t}^{\mathrm{H}}, \tag{5}$$

where $M_{f,t}^{(s)}$ denotes the time-frequency mask for the target signal.

In our previous work [5, 7, 11], the covariance matrix of the target signal is obtained by subtracting the covariance matrix of noise from that of the target plus noise signals. However, this subtraction sometimes means that the resultant matrix is not positive definite and makes the algorithm unstable. This often occurred especially with a frame-by-frame update in our preliminary experiments, therefore we simply use Eq. (5) in this work.

3. FRAME-BY-FRAME UPDATE RULE FOR MVDR BEAMFORMER

Let us assume we update the beamformer coefficients at every time frame, and the coefficients and the covariance matrices have a time index t. One possible approach for adaptive beamforming is to use an iterative algorithm such as the gradient descent algorithm described in [9, 12]. Unlike the iterative algorithm, we derive an update rule for the coefficients frame-by-frame based on the Woodbury matrix identity, which allows us to obtain the closed-form solution of the MVDR beamformer described in Eq. (4) at every time frame. This means that we achieve the global minimum value of the objective function described in Eq. (2) at every time frame.

To obtain the MVDR beamformer at time t, we compute the inverse matrix of $\mathbf{Y}_{f,t}$ based on the Woodbury matrix identity. A recursive update rule for the covariance matrix can be described as

$$\mathbf{Y}_{f,t} = \mathbf{Y}_{f,t-1} + \mathbf{y}_{f,t} \mathbf{y}_{f,t}^{\mathrm{H}}. \tag{6}$$

Note that we ignore the scalar scaling factor here since it is eventually canceled out in Eq. (4). The inverse of the covariance matrix at time t can be obtained based on Eq. (6) and the Woodbury matrix identity as

$$\mathbf{Y}_{f,t}^{-1} = \mathbf{Y}_{f,t-1}^{-1} - \frac{\mathbf{Y}_{f,t-1}^{-1} \mathbf{y}_{f,t} \mathbf{y}_{f,t}^{\mathrm{H}} \mathbf{Y}_{f,t-1}^{-1}}{(1 + \mathbf{y}_{f,t}^{\mathrm{H}} \mathbf{Y}_{f,t-1}^{-1} \mathbf{y}_{f,t})}. \tag{7}$$

By having the initial value of the inverse of the covariance matrix $\mathbf{Y}_{f,0}^{-1}$, we can compute the inverse matrix incrementally at every time frame without any additional inverse operation.

The covariance matrix of the target signal at time t can also be obtained recursively as

$$\mathbf{R}_{f,t}^{(s)} = \mathbf{R}_{f,t-1}^{(s)} + M_{f,t}^{(s)} \mathbf{y}_{f,t} \mathbf{y}_{f,t}^{\mathrm{H}}, \tag{8}$$

where we ignore the scaling factor for the same reason.

From Eqs. (7) and (8), the beamformer coefficients can be updated at every time frame by

$$\mathbf{w}_{f,t} = \frac{\mathbf{Y}_{f,t}^{-1} \mathbf{R}_{f,t}^{(s)} \mathbf{d}}{\mathrm{tr}(\mathbf{Y}_{f,t}^{-1} \mathbf{R}_{f,t}^{(s)})}. \tag{9}$$

This update rule achieves the global minimum of the objective function described in Eq. (2) with the entire observed signals obtained up to the current time frame.

4. MASK ESTIMATOR FOR FRAME-BY-FRAME BEAMFORMER

There are several ways to obtain time-frequency masks. One approach is to use a generative model, e.g., a complex Gaussian mixture model (CGMM) as in [5,7,11,13]. However, this generative model-based approach relies on us using the statistics of the observed signals to estimate the model parameters, and so this approach needs to aggregate a sufficient amount of observations for precise mask estimation. Hence, it is difficult to perform mask estimation frame-by-frame without a time delay.

On the other hand, neural network-based mask estimation has been investigated [4, 6, 8, 14–16], where time-frequency masks can be obtained as outputs of neural networks. Although BLSTMs have typically been used for mask estimation in many studies [4, 6, 14–16], we can also use unidirectional LSTMs instead of the BLSTMs to perform mask estimation frame-by-frame as in [8].

5. EXPERIMENTAL EVALUATION

We evaluated our frame-by-frame beamformer update on the CHiME-3 dataset [17]. We investigated ASR performance in terms of word error rates (WERs) with online and batch processing for both mask and beamformer estimation. The performance was also compared with that obtained with the baseline CGMM-based beamformer [7].

5.1. Data

We used the CHiME-3 dataset [17] for evaluation. Audio signals were recorded in four noisy public areas with six microphones attached to a tablet device. The noisy areas included a bus (BUS), café (CAF), pedestrian area (PED), and street junction (STR). The training data consisted of 7138 simulated noisy recordings and 1600 real noisy recordings. The simulated noisy data were generated by convolving impulse responses and clean speech signals from the WSJ0 corpus [18]. The development set consisted of 1640 simulated and 1640 real noisy recordings, which was used to tune hyperparameters. The evaluation set consisted of 1320 simulated and 1320 real noisy recordings and was used for performance evaluation. The sampling rate was 16 kHz. The CHiME-3 dataset is described in further detail in [17].

5.2. Settings

For frame-by-frame mask estimation, we used one LSTM layer followed by 2 fully-connected layers with ReLU activation functions and one fully-connected layer with a sigmoid activation function. The LSTM layer and the 2 ReLU layers had 256, 513 and 513 units, respectively. The final fully-connected layer projected dimensions from 513 to 201, which is the number of frequency bins. We performed a short-time Fourier transform with a 25 ms window length, an 10 ms window shift and a hanning window. As input features, we used log-magnitude spectra, which were averaged over the 6 channels. Optionally, we concatenated the input features at the 5 previous and 5 following frames, which introduced a time delay of 50 ms for feature extraction. The input features were normalized by using a cumulative moving average computed at every time frame. The LSTMs were trained on the simulated training dataset by minimizing the mean squared error between the log-magnitude spectra of clean signals and enhanced signals obtained by masking. The rmsprop algorithm [19] was used for model training, where the learning rates were set at $l = 0.00001$ for the model with no frame concatenation and at $l = 0.0001$ for that with an 11-frame concatenation, respectively. The learning rates were tuned by using the development set. The mini-batch size was set at 128, and the number of maximum epochs was set at 20. After the iteration, the best models were picked based on the loss on the development set. The initial value of the inverse of the covariance matrix $\mathbf{Y}_{f,0}^{-1}$ was set at an identity matrix.

For comparison, we used the CGMM-based beamformer with batch and block-wise processing [7]. The block size was set at 250 ms as in [7], which means we experienced a time delay of 250 ms before obtaining enhanced signals. Note that the MVDR beamformer used for the CGMM-based system was slightly different from the one we used for our neural network-based systems since it used subtraction to obtain the covariance matrix of the target signals as described in [7]. We used a subtraction-based covariance estimation for the baseline CGMM-based system so that we followed our previous work exactly. For additional investigation, we performed a beamformer estimation with an entire utterance (batch processing), block-wise processing and frame-by-frame processing. Moreover, we compared the result with BLSTM-based mask estimation to investigate the performance degradation caused by the frame-by-frame mask estimation based on the LSTMs. The BLSTMs were realized with the same number of parameters as the LSTMs.

For an ASR system, we used a deep convolutional neural network (CNN) acoustic model [20, 21] and a class-based recurrent neural network (RNN) language model [22, 23] as in our previous work [11]. The CNN acoustic model consisted of five convolution layers and two max-pooling layers, where all the layers contained 180 feature maps. The last convolution layer was followed by three fully-connected layers with 2048 units and a softmax layer with 5976 units. The units of the softmax layer corresponded to context-dependent HMM states. We used 10 classes for the RNN language model, which consisted of 500 hidden recurrent units. The recognizer was equivalent to the speaker-independent (SI) system in our CHiME-3 paper [11].

Table 1. WERs [%] obtained for the CHiME-3 development set.

systems	mask estimation	beamformer estimation	simu					real					total avg
			avg	BUS	CAF	PED	STR	avg	BUS	CAF	PED	STR	
unprocessed	-	-	8.24	6.42	8.48	7.51	10.56	9.01	6.03	14.00	8.05	7.94	8.62
baseline CGMM MVDR	batch	batch	**5.19**	5.15	6.39	4.45	4.78	**5.00**	7.82	3.85	3.92	4.39	**5.09**
	block-wise (250 ms)	block-wise (250 ms)	5.57	5.56	6.59	4.56	5.58	5.35	8.28	4.44	4.19	4.50	5.46
BLSTM MVDR	batch	batch	5.37	5.43	6.76	4.28	4.99	6.21	9.74	5.10	4.53	5.49	5.79
proposed LSTM MVDR	frame-by-frame	batch	5.74	5.78	7.61	4.47	5.10	6.54	10.09	5.59	4.71	5.77	6.14
	frame-by-frame	block-wise (250 ms)	6.14	6.12	7.76	4.82	5.86	6.94	10.55	5.90	4.82	6.47	6.54
	frame-by-frame	frame-by-frame	6.22	6.14	7.99	4.97	5.78	7.13	10.89	6.02	4.99	6.61	6.67
proposed LSTM MVDR + 11 frame concatenation	frame-by-frame	batch	5.43	5.53	7.01	4.40	4.79	6.12	9.35	5.18	4.59	5.35	5.77
	frame-by-frame	block-wise (250 ms)	5.85	6.02	7.36	4.66	5.37	6.53	9.79	5.56	4.75	6.02	6.19
	frame-by-frame	frame-by-frame	5.95	6.14	7.39	4.63	5.65	6.83	10.33	5.83	4.96	6.19	6.39

Table 2. WERs [%] obtained for the CHiME-3 evaluation set.

systems	mask estimation	beamformer estimation	simu					real					total avg
			avg	BUS	CAF	PED	STR	avg	BUS	CAF	PED	STR	
unprocessed	-	-	10.17	8.37	11.69	9.86	10.78	15.60	22.55	16.21	12.89	10.74	12.89
baseline CGMM MVDR	batch	batch	7.90	5.40	7.34	9.34	9.53	**8.37**	11.44	6.97	8.43	6.65	8.14
	block-wise (250 ms)	block-wise (250 ms)	8.47	6.85	8.39	8.39	10.25	9.13	12.96	8.27	7.85	7.43	8.80
BLSTM MVDR	batch	batch	**6.87**	5.92	7.68	6.56	7.30	9.33	13.83	8.76	6.88	7.87	**8.10**
proposed LSTM MVDR	frame-by-frame	batch	7.51	6.07	9.00	7.40	7.56	10.13	14.08	10.85	7.40	8.19	8.82
	frame-by-frame	block-wise (250 ms)	8.03	6.28	8.98	8.14	8.72	10.90	15.16	11.36	8.43	8.65	9.46
	frame-by-frame	frame-by-frame	8.22	6.50	9.15	8.50	8.72	10.91	15.63	11.19	8.31	8.50	9.56
proposed LSTM MVDR + 11 frame concatenation	frame-by-frame	batch	7.29	5.83	8.61	7.25	7.47	9.59	13.03	9.56	7.42	8.34	8.44
	frame-by-frame	block-wise (250 ms)	7.61	5.75	8.69	7.73	8.26	10.42	13.82	10.57	8.13	9.16	9.01
	frame-by-frame	frame-by-frame	7.83	6.07	8.74	7.92	8.59	10.54	14.53	10.68	8.03	8.91	9.18

5.3. Results

Tables 1 and 2 show the WERs obtained for the development and evaluation sets, respectively. With frame-by-frame mask and beamformer estimation, our LSTM-based MVDR beamformer achieved a large performance gain from unprocessed signals with a short time delay. Furthermore, 11-frame concatenation helped to improve ASR performance with the LSTM-based beamformer. Our LSTM-based frame-by-frame beamformer with and without frame concatenation outperformed a baseline CGMM-based block-wise beamformer on the simulated evaluation set even with a shorter time delay. Compared with batch processing, as expected, we experienced a slight performance degradation when using the LSTM-based frame-by-frame mask estimator instead of the BLSTM-based mask estimator.

5.4. Discussion and future work

One possible reason for the (B)LSTM-based beamformer working better with the simulated dataset is that the (B)LSTMs were trained on simulated parallel training data. This limitation will be removed by performing end-to-end optimization as in [8, 14–16] and/or recently-proposed adversarial training for mask estimators [24]. These approaches allow us to train a (B)LSTM-based mask estimator with real noisy recordings without corresponding clean signals. Combining these approaches with our LSTM-based frame-by-frame beamformer will constitute our future work.

6. CONCLUSION

In this paper, we proposed a frame-by-frame update rule for a mask-based MVDR beamformer. The update rule was combined with frame-by-frame mask estimation based on the LSTMs, which enabled us to perform MVDR beamforming with frame-by-frame processing. The proposed algorithm achieved the global minimum of the objective function for the MVDR beamformer at every time frame without any matrix inversion as a result of the Woodbury matrix identity. Experimental results showed that our frame-by-frame MVDR beamformer outperformed block-wise CGMM-based MVDR beamforming on the CHiME-3 simulated dataset even with a shorter time delay.

7. REFERENCES

[1] B. Li et al., “Acoustic modeling for google home,” *INTERSPEECH*, 2017.

[2] X. Xiao et al., “Deep beamforming networks for multi-channel speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5745–5749.

[3] B. Li et al., “Neural network adaptive beamforming for robust multichannel speech recognition.,” in *INTERSPEECH*, 2016, pp. 1976–1980.

[4] J. Heymann et al., “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.

[5] T. Higuchi et al., “Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.* IEEE, 2016, pp. 5210–5214.

[6] H. Erdogan et al., “Improved MVDR beamforming using single-channel mask prediction networks,” in *INTERSPEECH*, 2016.

[7] T. Higuchi et al., “Online MVDR beamformer based on complex Gaussian mixture model with spatial prior for noise robust ASR,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 780–793, 2017.

[8] X. Xiao et al., “On time-frequency mask estimation for MVDR beamforming with application in robust speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3246–3250.

[9] O. L. Frost, “An algorithm for linearly constrained adaptive array processing,” *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, 1972.

[10] M. Souden et al., “A multichannel MMSE-based framework for speech source separation and noise reduction,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 9, pp. 1913–1928, 2013.

[11] T. Yoshioka et al., “The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices,” in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 436–443.

[12] R. Haeb-Umbach and E. Warsitz, “Adaptive filter-and-sum beamforming in spatially correlated noise,” in *Proc. IWAENC*, 2005, pp. 125–128.

[13] J. Du et al., “The USTC–iFlytek system for CHiME-4 challenge,” *Proc. CHiME*, pp. 36–38, 2016.

[14] J. Heymann et al., “BEAMNET: End-to-end training of a beamformer-supported multi-channel ASR system,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5325–5329.

[15] T. Ochiai et al., “Multichannel end-to-end speech recognition,” *International Conference on Machine Learning (ICML)*, 2017.

[16] T. Ochiai et al., “Does speech enhancementwork with end-to-end ASR objectives?: Experimental analysis of multichannel end-to-end ASR,” *Machine Learning for Signal Processing (MLSP)*, 2017.

[17] J. Barker et al., “The third ’CHiME’ speech separation and recognition challenge: Dataset, task and baselines,” in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 504–511.

[18] 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*. Association for Computational Linguistics, 1992, pp. 357–362.

[19] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, 2012.

[20] O. Abdel-Hamid et al., “Convolutional neural networks for speech recognition,” *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[21] T. N. Sainath et al., “Deep convolutional neural networks for large-scale speech tasks,” *Neural Networks*, vol. 64, pp. 39–48, 2015.

[22] T. Mikolov et al., “Recurrent neural network based language model,” in *Interspeech*, 2010, pp. 1045–1048.

[23] T. Mikolov et al., “Empirical evaluation and combination of advanced language modeling techniques,” in *Interspeech*, 2011, pp. 605–608.

[24] T. Higuchi et al., “Adversarial training for data-driven speech enhancement without parallel corpus,” in *The 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2017)*, to appear, 2017.