ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414442

# ON LOSS FUNCTIONS FOR DEEP-LEARNING BASED T60 ESTIMATION

*Yuying Li†, Yuchen Liu‡ and Donald S. Williamson‡*

†Department of Intelligent Systems Engineering, Indiana University, USA
‡Department of Computer Science, Indiana University, USA
{liyuy,liu477,williads}@indiana.edu

## ABSTRACT

Reverberation time, $T_{60}$, directly influences the amount of reverberation in a signal, and its direct estimation may help with dereverberation. Traditionally, $T_{60}$ estimation has been done using signal processing or probabilistic approaches, until recently where deep-learning approaches have been developed. Unfortunately, the appropriate loss function for training the network has not been adequately determined. In this paper, we propose a composite classification- and regression-based cost function for training a deep neural network that predicts $T_{60}$ for a variety of reverberant signals. We investigate pure-classification, pure-regression, and combined classification-regression based loss functions, where we additionally incorporate computational measures of success. Our results reveal that our composite loss function leads to the best performance as compared to other loss functions and comparison approaches. We also show that this combined loss function helps with generalization.

***Index Terms***— T60 estimation, reverberation time estimation, deep neural networks, loss function

## 1. INTRODUCTION

Reverberation is a natural phenomenon that occurs when sounds from a source reflect off of different surfaces before it reaches a microphone or person's ears. Reverberation has been shown to be costly to many speech-based applications, e.g. speech enhancement [1, 2], automatic speech recognition [3, 4], and speaker localization [5, 6], to name a few. This occurs because the reverberation may cause a signal to smear across time and frequency.

The amount of reverberation influences descriptors like $T_{60}$ and the direct speech to reverberation ratio (DRR). DRR is a logarithmic energy ratio between the direct and reverberant components of a signal, where higher numbers signify less reverberation. $T_{60}$ tells how long it takes a given signal to decay by 60 dB. In this case, higher $T_{60}$ (or reverberation) times indicate more reverberation. Reasonable estimates of these two parameters convey meaningful information about the room environment, and they also disclose information about the corresponding room impulse response. Hence, adequately estimating them may help with auditory scene analysis [7] and dereverberation [8]. This paper focuses on $T_{60}$ estimation.

$T_{60}$ estimation has traditionally been accomplished using signal processing techniques. In [9], the author compares two different methods for extracting room acoustic parameters from reverberated speech. The first method uses statistical machine learning. The second method produces a maximum likelihood estimate on decay phases at the end of the utterances. The second method is also extended by estimating parameters related to the balance of early and late energies in the impulse response. The authors in [10] propose an algorithm for blind estimation of reverberation in speech signals by applying a spectral decomposition on reverberation signals. Partial reverberation time estimates are determined in all signal subbands.

Like many other problems, $T_{60}$ estimation is now being investigated using deep neural networks (DNN)s. In [11], the authors propose a multi-layer perception (MLP) approach to $T_{60}$ estimation. More specifically, they extract features using a Gabor filterbank, and supply the features as inputs to a MLP. The input consists of the Gabor response, which has nine frames of the Gabor feature vector. This is a frame-level prediction approach, where the single $T_{60}$ for the utterance is predicted at each time frame. A decision rule is then used to generate a utterance-level estimate from the frame-level estimates. An updated version of this approach is presented in [12], where this approach jointly estimates reverberation time and DRR. In [13], the authors propose a fully-connected convolutional neural network (CNN), where the output of each layer are downsampled, until the last layer outputs a single value, which estimates $T_{60}$. Hence, a single prediction is made at the utterance level. This approach is extended in [14], where the CNN is modified and data is augmented to further improve performance.

The above approaches mostly use the mean-square error (MSE) as the loss function to estimate $T_{60}$. Since other speech-related tasks have shown that the MSE is sub-optimal, alternative loss functions for reverberation time estimation should be explored. Likewise, work in ASR [15] and speech assessment [16] have shown that treating speech-tasks as classification, rather than regression problems is beneficial.

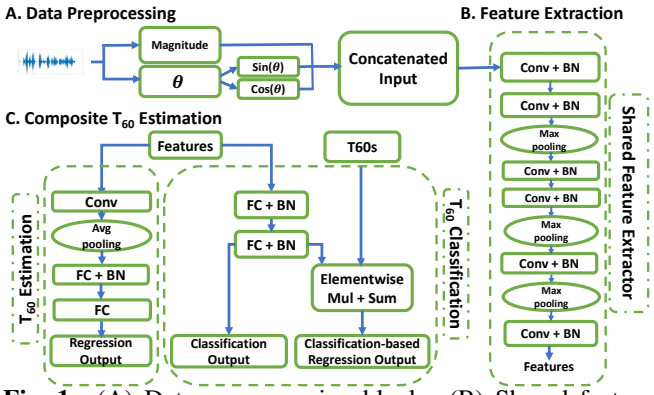


**Fig. 1**. (A) Data preprocessing block. (B) Shared feature extraction block. (C) Composite $T_{60}$ estimation block.

In this paper, we propose a composite classification and regression based loss function for estimating reverberation time for a variety of seen and unseen reverberant conditions. In particular, we explore a multi-task framework that uses magnitude and phase features of the signals, incorporates an additional convolutional-based feature extraction stage, and generates predictions using regression, classification, and classification-based regression training targets. Standard classification (e.g. cross entropy) and regression (MSE) loss functions are also combined with standard measures of performance (e.g. mean-absolute error, Pearson's correlation coefficient, and Spearman's rank correlation) to form a composite loss function.

## 2. PROPOSED APPROACH

The proposed approach for estimating $T_{60}$ directly from a reverberant signal is shown in Fig. 1. The approach performs data preprocessing, shared feature extraction, and composite $T_{60}$ estimation.

### 2.1. Data pre-processing

Reverberant speech, $x(t)$, can be generated by convolving anechoic speech, $s(t)$, with a room impulse response (RIR)

$$x(t) = s(t) * h(t) \tag{1}$$

where * denotes convolution and $h(t)$ is the RIR. Given a reverberant speech signal in the time domain, the data preprocessing block (Fig.1A) computes the short-time Fourier transform (STFT) of the signal and returns the log-magnitude and phase response, which are both two-dimensional matrices. From the phase information, we further compute the $\sin\theta$ and $\cos\theta$ instead of using phase directly, where $\theta$ is the phase angle of the STFT matrix. This is done since prior studies have shown that this phase information is useful for deep-learning based speech problems [17, 18, 19, 20]. The features are subsequently concatenated along time to form a combined feature vector. We then normalize the feature vector so that it has zero mean and unit variance for each frequency channel.

### 2.2. Feature Extraction and T60 estimation

The feature extraction network structure we propose is depicted in Fig.1B. The pre-processed input matrix passes to a shared feature extractor, which consists of six 2D convolutional layers (Conv) with rectified linear unit (ReLU) activation functions. Batch normalization (BN) is applied after each convolutional layer [21]. The number of kernel filters is set as follows: 16 kernel filters for the first two layers, 32 kernel filters for the middle two layers and 64 kernel filters for the last two layers. Max pooling is performed after the first two Conv layers, the middle two Conv layers, and between the last two Conv layers. The kernel sizes of the maxpooling layers are set to $2 \times 2$. Note that this architecture is based on [16], since it performed well for a similar but different speech assessment task. We did, however, make modifications as discussed next.

Next, the output from the shared feature extractor is simultaneously applied to two related tasks (Fig.1C). The left portion of the network is for the regression task, which directly estimates $T_{60}$. It passes the inputted feature to one Conv layer with a ReLU activation function, and follows with a 2D average pooling layer that outputs a scalar value. 128 kernel filters are used, where the kernel size is $3 \times 3$. One fully connected layer (FC) with batch normalization follows, where a leaky ReLU activation function is used. Its negative slope is set to 0.1. Finally one more FC layer is applied with a ReLU activation function as the final layer for the $T_{60}$ estimation. The ReLU activation ensures that the estimated $T_{60}$ is positive valued.

The right portion of the network shown in Fig.1C is used for the classification task, which we further decompose into two sub-tasks. The first sub-task aims to predict a one-hot vector, and results in the predicted probabilities of each $T_{60}$ class, $C_{out}$. The second sub-task generates a regression-based output from the predicted probabilities. More specifically, we compute the weighted sum between a vector of the reverberation times and the classification probabilities $C_{out}$. This classification-based regressed estimate is computed as shown in Eq. (2).

$$CReg_{T_{60}} = \sum_{i=1}^{H} \left( C_{out}^{i} \times T_i \right), i = 1, \cdots, H \tag{2}$$

where $\times$ denotes point-wise multiplication, $H$ denotes the number of classes, $T_i$ is the $T_{60}$ time of the $i$-th class, $C_{out}^{i}$ is the estimated class-probability for the $i$-th class, and $CReg_{T_{60}}$ denotes the classification-based regression output for the current signal. For this second task, we pass the extracted feature from the shared feature extractor to the classification part of network, which consists of two FC layers with leaky ReLU and softmax activation functions are applied to the last layer. Batch normalization is also applied.

### 2.3. Cost Function

Four cost functions are proposed and used to train the above network. First, we propose to combine the cross-entropy loss $L_{cel}$ from the classification sub-task and mean-squared error (MSE) $L_{reg}$ from the regression part, to update the weight matrices. In other words, we do not generate a classification-based regression output. The loss function for this approach is shown below.

$$L^A_{total} = \beta * L_{cel} + (1 - \beta) * L_{reg} \tag{3}$$

where $\beta \in [0, 1]$ controls the weight between the cross-entropy and regression based loss functions. This loss function is denoted as $L^A_{total}$.

Since we separate the classification task into two sub-tasks, we also include both the cross-entropy loss, $L_{cel}$, and the classification-based regression loss, $L_{creg}$, into the classification and total loss functions. The MSE is used for the classification-based regression loss, $L_{creg}$.

$$\begin{aligned} L^B_{total} =& \beta * (\alpha * L_{cel} + (1 - \alpha) * L_{creg}) + \\ & (1 - \beta) * L_{reg} \end{aligned} \tag{4}$$

Here $\alpha \in [0, 1]$ controls the weight between the cross-entropy and classification-based regression loss functions of the classification sub-task. We denote this loss function as $L^B_{total}$.

From previous research [22, 23], evaluation scores such as the scale-invariant source-to-noise ratio (SI-SNR) are used within the cost functions to update the weight matrices for different speech enhancement tasks. In a similar fashion, we propose to add Pearson's correlation coefficient (PCC) and Spearman's rank correlation coefficient (SRCC) as additional components of our composite cost function. This loss function is denoted as $L^C_{total}$, and is shown below

$$L^C_{total} = L^B_{total} - |\rho_{reg}| - |\eta_{reg}| - |\rho_{cls}| - |\eta_{cls}|, \tag{5}$$

where $\rho_{reg}$ and $\eta_{reg}$ denote the PCC and SRCC scores for the regression task, and $\rho_{cls}$ and $\eta_{cls}$ denote the PCC and SRCC scores for classification-based regression output. Here, $|\cdot|$ denotes absolute value. Since we want to minimize total loss, we hence want to maximize the absolute value of each correlation in order to ensure a high correlation between the target and estimates.

Our fourth loss function incorporates the mean absolute error (MAE) from the regression part, to determine if this additional term impacts performance, since it is a standard metric for reverberation time estimation.

$$\begin{aligned} L^D_{total} =& \beta * (\alpha * L_{cel} + (1 - \alpha) * (L_{creg} + M_{creg})) \\ & + (1 - \beta) * (L_{reg} + M_{reg}) \\ & - |\rho_{reg}| - |\eta_{reg}| - |\rho_{cls}| - |\eta_{cls}| \end{aligned} \tag{6}$$

where $M_{creg}$ denotes the MAE of classification-based regression subnet and $M_{reg}$ denotes the MAE of the regression subnet.

All the above models with different cost functions share the same set of parameters: batch size is 50, Adam optimization is applied, and the learning rate is set to 0.001. All the models are trained using the standard back propagation algorithm for 100 epochs. Note that we experimented with different cost functions (e.g. different combination of $\beta$ and $\alpha$), but we empirically determined the best combination of $\beta$ and $\alpha$ for the different approaches.

## 3. EXPERIMENTS

### 3.1. Experimental Setup

The proposed system is evaluated using the TIMIT corpus [24], which contains various native English speakers from different regions of the United States. In our experiments, we randomly select 5000, 500, and 500 utterances to construct our training, validation and testing datasets. All 6000 utterances are downsampled to 8kHz. We simulate RIRs from 11 different rooms using the imaging method [25], with dimensions of: $9\text{m} \times 8\text{m} \times 7\text{m}$, $10\text{m} \times 7\text{m} \times 3\text{m}$, $6\text{m} \times 6\text{m} \times 10\text{m}$, $8\text{m} \times 10\text{m} \times 4\text{m}$, $7\text{m} \times 7\text{m} \times 8\text{m}$, $7\text{m} \times 9\text{m} \times 5\text{m}$, $8\text{m} \times 8\text{m} \times 10\text{m}$, $10\text{m} \times 10\text{m} \times 8\text{m}$, $8\text{m} \times 8\text{m} \times 6\text{m}$, $7\text{m} \times 8\text{m} \times 6\text{m}$ and $9\text{m} \times 9\text{m} \times 10\text{m}$. The distance between the receiver and the speaker is set to $1m$ in all cases. We select 13 different reverberation times from 0.3s to 1.5s, with steps of 0.1s. We simulated 500 different RIRs for each $T_{60}$ in the first 10 rooms that are used to generate the training set, another 100 RIRs for each $T_{60}$ in the same room settings to separately generate the validation and seen room testing sets, and 500 different RIRs for each $T_{60}$ in the 11th room for unseen testing. We convolve each RIR with one utterance for each $T_{60}$. As a result, there are $5000 \times 13 (T_{60}\text{s})$ = 65,000 reverberant utterances in the training set; $500 \times 13 (T_{60}\text{s})$ = 6,500 reverberant utterances in the validation, seen and unseen testing sets.

While the length of the $h(t)$ can vary due to $T_{60}$, we pad zeros to $h(t)$ before the convolution, which makes the length of $h(t)$ the same. Also, we cut all the clean signals to 6 seconds, and then convolve the clean signals with the padded RIRs.

The short-time Fourier transform (STFT) is computed using a 480-sample Hamming window, 512-point fast Fourier transform (FFT), and 75% overlap between successive frames.

## 4. EVALUATION

### 4.1. Comparison Approach

In our experiments, we compare our proposed system with two baseline approaches: $T_{60}$ estimation using convolutional neural network (CNN) [14] and $T_{60}$ estimation based on spectro-temporal modulation filtering [11]. Note that data augmentation is not applied when training the CNN approach from [14]. The feature we use for the CNN approach is the

**Table 1**. Seen Rooms Comparison with different approaches. $L_{total}(\cdot)$ denotes the loss function used for proposed system

| | MSE | | MAE | | $\rho$ | | $\eta$ | |
|---|---|---|---|---|---|---|---|---|
| | Reg | Cls | Reg | Cls | Reg | Cls | Reg | Cls |
| MLP [11] | 0.075 | – | 0.211 | – | 0.783 | – | 0.788 | – |
| CNN [14] | **0.044** | – | **0.196** | – | 0.931 | – | 0.940 | – |
| $\mathrm{L}^{A}_{total}(\beta = 0)$ | 0.057 | 0.145 | 0.208 | 0.329 | 0.929 | -0.128 | 0.939 | -0.107 |
| $\mathrm{L}^{A}_{total}(\beta = 0.4)$ | 0.270 | 0.033 | 0.425 | 0.147 | -0.211 | 0.927 | -0.165 | 0.940 |
| $\mathrm{L}^{A}_{total}(\beta = 1)$ | 0.448 | 0.198 | 0.566 | 0.365 | 0.092 | 0.120 | 0.101 | -0.013 |
| $\mathrm{L}^{B}_{total}(\beta = 0.3, \alpha = 0.1)$ | 0.176 | 0.135 | 0.347 | 0.318 | 0.781 | 0.573 | 0.819 | 0.635 |
| $\mathrm{L}^{C}_{total}(\beta = 0.4, \alpha = 0.2)$ | 0.131 | **0.022** | 0.289 | **0.116** | 0.609 | 0.955 | 0.606 | **0.973** |
| $\mathrm{L}^{D}_{total}(\beta = 0.3, \alpha = 0)$ | 0.098 | 0.093 | 0.270 | 0.228 | 0.771 | 0.808 | 0.800 | 0.816 |
| $\mathrm{L}^{D}_{total}(\beta = 0.9, \alpha = 0.1)$ | 0.120 | 0.057 | 0.290 | 0.204 | **0.955** | **0.963** | **0.958** | 0.968 |
| $\mathrm{L}^{D}_{total}(\beta = 0.3, \alpha = 1)$ | 0.284 | 0.250 | 0.435 | 0.412 | -0.013 | 0.428 | 0.003 | 0.430 |

**Table 2**. Unseen Room Comparison with different approaches. $L_{total}(\cdot)$ denotes the loss function used for proposed system

| | MSE | | MAE | | $\rho$ | | $\eta$ | |
|---|---|---|---|---|---|---|---|---|
| | Reg | Cls | Reg | Cls | Reg | Cls | Reg | Cls |
| MLP [11] | 0.092 | – | 0.239 | – | 0.715 | – | 0.723 | – |
| CNN [14] | 0.096 | – | 0.212 | – | 0.856 | – | 0.860 | – |
| $\mathrm{L}^{A}_{total}(\beta = 0)$ | **0.047** | 0.145 | **0.189** | 0.329 | 0.942 | -0.098 | 0.953 | -0.084 |
| $\mathrm{L}^{A}_{total}(\beta = 0.4)$ | 0.298 | 0.056 | 0.449 | 0.171 | -0.042 | 0.919 | -0.198 | 0.942 |
| $\mathrm{L}^{A}_{total}(\beta = 1)$ | 0.467 | 0.201 | 0.577 | 0.368 | 0.040 | 0.069 | 0.050 | -0.070 |
| $\mathrm{L}^{B}_{total}(\beta = 0.3, \alpha = 0.1)$ | 0.174 | 0.136 | 0.345 | 0.319 | 0.830 | 0.476 | 0.872 | 0.546 |
| $\mathrm{L}^{C}_{total}(\beta = 0.4, \alpha = 0.2)$ | 0.117 | **0.023** | 0.273 | **0.114** | 0.532 | 0.968 | 0.525 | **0.984** |
| $\mathrm{L}^{D}_{total}(\beta = 0.3, \alpha = 0)$ | 0.092 | 0.089 | 0.261 | 0.221 | 0.845 | 0.814 | 0.866 | 0.837 |
| $\mathrm{L}^{D}_{total}(\beta = 0.9, \alpha = 0.1)$ | 0.102 | 0.045 | 0.263 | 0.180 | **0.962** | **0.973** | **0.962** | 0.977 |
| $\mathrm{L}^{D}_{total}(\beta = 0.3, \alpha = 1)$ | 0.295 | 0.242 | 0.444 | 0.405 | 0.219 | 0.601 | 0.229 | 0.622 |

log-mel spectrogram. The architecture of the CNN matches what is reported in their paper. For the spectro-temporal modulation approach, the feature is extracted by using Gabor 2D filters. The features are then inputted into a 3-layer MLP [11], which is done in the original approach.

To evaluate the performance of the proposed system compared with comparison approaches, the MSE, MAE, PCC ($\rho$) and SRCC ($\eta$) are calculated. The MSE and MAE have no specific range, but for all cases, lower values indicate better performance. The PCC and SRCC have ranges from -1 to 1, where scores closer to 1 indicate better results.

Table 1 lists all scores for different approaches in seen rooms. The best performance is shown in bold for each metric. Comparing our proposed system with the regression only baseline models (MLP and CNN), the CNN has a better performance on regression task in terms of MSE and MAE as shown in the first and third columns of table 1. However, $L^{C}_{total}(\beta = 0.4, \alpha = 0.2)$ (composite without MAE) gives the best performance in terms of MSE and MAE across both regression and classification-based subnets. $L^{D}_{total}(\beta = 0.9, \alpha = 0.1)$ (composite with MAE) gives the best performance in terms of PCC across both regression and classification subnets, and it also performs well according to SRCC. For SRCC, $L^{C}_{total}$ shows the best performance overall. Overall, the best performing proposed approach clearly outperforms MLP and CNN approaches across all metrics. This also occurs when the objective measures (MAE, PCC and SRCC) are not included in the cost function $L^{A}_{total}$, indicating that the classification and regression subnets regularize each other to improve performance.

Table 2 lists all comparison scores for different approaches in the unseen room. The pure regression model in $L^{A}_{total}(\beta = 0)$ shows the best performance in terms of MSE and MAE for the regression task in the first and third columns of the table. In terms of MSE and MAE, $L^{C}_{total}$ (composite without MAE) performs the best overall according to MSE and MAE. $L^{D}_{total}$ (composite with MAE) with $\beta = 0.9, \alpha = 0.1$ performs best on both regression and classification-based regression subnets according to PCC. Overall, our proposed system outperforms CNN and MLP in terms of all scores in an unseen room, indicate its ability to generalize.

## 5. CONCLUSION

In this paper, we propose a composite classification and regression-based cost function for training a deep neural network that predicts $\mathrm{T}_{60}$. Our approach is different from recent methods and benefits from the two tasks. The results shows that the tradeoff between weighting classification versus regression tasks does influence results. Our approach also benefits from dividing the classification tasks into two subtasks. In the future, we would like to address real reverberant speech, and also determine better ways to tune $\alpha$ and $\beta$.

## 6. REFERENCES


[1] D. S. Williamson and D. L. Wang, "Speech dereverberation and denoising using complex ratio masks," in *Proc. ICASSP*. IEEE, 2017, pp. 5590–5594.

[2] T. Yoshioka, T. Nakatani, and M. Miyoshi, "Integrated speech enhancement method using noise suppression and dereverberation," *IEEE/ACM TASLP*, pp. 231–246, 2009.

[3] K. Kinoshita, M. Delcroix, and T. Yoshioka, "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proc. WASPAA*, 2013, pp. 1–4.

[4] R. Giri, M. L. Seltzer, and D. Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," in *Proc. ICASSP*, 2015, pp. 5014–5018.

[5] S. Chakrabarty and E Habets, "Broadband doa estimation using convolutional neural networks trained with noise signals," in *Proc. WASPAA*, 2017, pp. 136–140.

[6] H. Wang and J. Lu, "A robust doa estimation method for a linear microphone array under reverberant and noisy environments," *arXiv preprint arXiv:1904.06648*, 2019.

[7] Y. Hioka, K. Niwa, S. Sakauchi, and Y Haneda, "Estimating direct-to-reverberant energy ratio using d/r spatial correlation matrix model," *IEEE/ACM TASLP*, pp. 2374–2384, 2011.

[8] N. Kilis and N. Mitianoudis, "A novel scheme for single-channel speech dereverberation," *Acoustics*, vol. 1, no. 3, pp. 711–725, 2019.

[9] P. Kendrick, T. J. Cox, F. F. Li, Y. Zhang, and J. A. Chambers, "Monaural room acoustic parameters from music and speech," *JASA*, vol. 124, no. 1, pp. 278–287, 2008.

[10] T. D. M. Prego, A. A. de Lima, R. Zambrano-López, and S. L. Netto, "Blind estimators for reverberation time and direct-to-reverberant energy ratio using subband speech decomposition," in *Proc. WASPAA*. IEEE, 2015, pp. 1–5.

[11] F. Xiong, S. Goetze, and Meyer B. T., "Blind estimation of reverberation time based on spectro-temporal modulation filtering," in *Proc. ICASSP*. IEEE, 2013, pp. 443–447.

[12] F. Xiong, S. Goetze, and B. T. Meyer, "Joint estimation of reverberation time and direct-to-reverberation ratio from speech using auditory-inspired features," *arXiv preprint arXiv:1510.04620*, 2015.

[13] H. Gamper and I. J. Tashev, "Blind reverberation time estimation using a convolutional neural network," in *Proc. IWAENC*. IEEE, 2018, pp. 136–140.

[14] N. J. Bryan, "Impulse response data augmentation and deep neural networks for blind room acoustic parameter estimation," in *Proc. ICASSP*. IEEE, 2020, pp. 1–5.

[15] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*, 2014, pp. 1764–1772.

[16] X. Dong and D. S. Williamson, "A classification-aided framework for non-intrusive speech quality assessment," in *Proc. WASPAA*. IEEE, 2019, pp. 100–104.

[17] Z-Q. Wang and D. L. Wang, "Combining spectral and spatial features for deep leaerning based blind speaker separation," *IEEE/ACM TASLP*, pp. 457–468, 2018.

[18] J. Lee and H-G. Kang, "A joint learn-ing algorithm for complex-valued tf masks in deeplearning-based single-channel speech enhancement sys-tems," *IEEE/ACM TASLP*, pp. 1098–1108, 2019.

[19] Z. Zhang, D. S Williamson, and Y. Shen, "Investigation of phase distortion on perceived speech quality for hearing-impaired listeners," *arXiv preprint arXiv:2007.14986*, 2020.

[20] Y. Li and D. S Williamson, "A return to dereverberation in the frequency domain using a joint learning approach," in *Proc. ICASSP*. IEEE, 2020, pp. 7549–7553.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reduing internal covariate shift," in *arXiv preprint arXiv:1502.03167*, 2015.

[22] Y. Luo and Mesgarani N., "Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM TASLP*, pp. 1256–1266, 2019.

[23] Y. Luo and Mesgarani N., "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *Proc. ICASSP*. IEEE, 2018, pp. 696–700.

[24] J. S. "Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. " Dahlgren, "Darpa timit acoustic phonetic continuous speech corpus," in *Available: http://www.ldc.upenn.edu/Catalog/LDC93S1.html*, 1993.

[25] E. Habets, "Room impulse response generator (http://home.tiscali.nl/ehabets/rir generator.html)," 2010.