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/ICASSP39728.2021.9414464

END-TO-END DEREVERBERATION, BEAMFORMING, AND SPEECH RECOGNITION WITH IMPROVED NUMERICAL STABILITY AND ADVANCED FRONTEND

Wangyou Zhang[1], *Christoph Boeddeker*[2], *Shinji Watanabe*[3], *Tomohiro Nakatani*[4],
Marc Delcroix[4], *Keisuke Kinoshita*[4], *Tsubasa Ochiai*[4], *Naoyuki Kamo*[4],
Reinhold Haeb-Umbach[2], *Yanmin Qian*[1]
[1]MoE Key Lab of Artificial Intelligence, AI Institute, SpeechLab, Shanghai Jiao Tong University, China
[2]Paderborn University, Germany [3]Johns Hopkins University, USA [4]NTT Corporation, Japan

ABSTRACT

Recently, the end-to-end approach has been successfully applied to multi-speaker speech separation and recognition in both single-channel and multichannel conditions. However, severe performance degradation is still observed in the reverberant and noisy scenarios, and there is still a large performance gap between anechoic and reverberant conditions. In this work, we focus on the multichannel multi-speaker reverberant condition, and propose to extend our previous framework for end-to-end dereverberation, beamforming, and speech recognition with improved numerical stability and advanced frontend subnetworks including voice activity detection like masks. The techniques significantly stabilize the end-to-end training process. The experiments on the spatialized wsj1-2mix corpus show that the proposed system achieves about 35% WER relative reduction compared to our conventional multi-channel E2E ASR system, and also obtains decent speech dereverberation and separation performance (SDR = 12.5 dB) in the reverberant multi-speaker condition while trained only with the ASR criterion.

Index Terms— Neural beamformer, overlapped speech recognition, dereverberation, speech separation, cocktail party problem

1. INTRODUCTION

With the development of deep learning, much progress has been achieved in the speech processing field, including both speech enhancement [1–3] in the frontend and automatic speech recognition (ASR) [4–6] in the backend. In recent years, more and more interests have been focused on the deep learning based speech processing in the cocktail party scenario [7,8]. In this scenario, there are usually multiple speakers talking simultaneously, even with the presence of background noise and reverberation. It is much more difficult to cope with than in the clean and anechoic conditions, and the ASR performance is still far behind humans in such conditions.

In the cocktail party scenario, while it is straightforward to combine separately trained speech enhancement and speech recognition components as one system, as investigated in many prior studies [9, 10], the end-to-end (E2E) optimization of all involved components is also an important and interesting research topic. The E2E system can naturally reduce the mismatch between different components through joint training. In addition, only the noisy signal and the corresponding transcriptions are required for the E2E training of both frontend and backend, making it much easier for data collection and model training in real applications. Some prior work has illustrated the potential of E2E optimized systems. Settle et al. [11] proposed a joint training framework, combining the chimera++ network [12] and end-to-end ASR [13] for single-channel multi-speaker speech separation and recognition. In the multichannel condition, the neural beamformer [14, 15] based speech enhancement is often applied to better utilize the spatial information. *(1) Single-speaker cases:* In [16–18], the neural beamformer is jointly trained with the acoustic / end-to-end ASR model for denoising and speech recognition. Subramanian et al. [19] further included dereverberation in the joint training, which is based on the weighted prediction error (WPE) [20] algorithm. *(2) Multi-speaker cases:* Chang et al. [21] proposed the MIMO-Speech architecture, where the beamformer is jointly trained with ASR to perform speech separation.

In this paper, we aim to build a robust framework for the fully end-to-end optimization of dereverberation, beamforming (denoising and separation), and speech recognition. In our prior work [22], some preliminary attempts have been made to explore the end-to-end training of three components: WPE-based dereverberation, neural beamforming, and end-to-end ASR. However, the well-known numerical instability issue [23] in operations of both WPE and beamforming, usually caused by the singularity in the matrix inverse operation, is still unsolved in [22], leading to performance degradation or even misleading the model convergence.

In this work, we try to tackle this problem, by proposing four techniques to improve the stability and performance of the end-to-end system. These methods have been proven extremely helpful in our setup, significantly mitigating the numerical instability issue during training. Based on these techniques, we propose a robust architecture that supports the end-to-end training of different beamformer variants and ASR, which are also compared in our experiments. In addition, the voice activity detection (VAD) like mask [19, 24] for WPE and beamforming is introduced to mitigate the frequency permutation problem in the end-to-end training, as described in Section 2.3. Our experiments on the spatialized wsj1-2mix [21] corpus show that the proposed approaches can achieve significant performance improvement compared to the previous system.

2. END-TO-END FRAMEWORK FOR DEREVERBERATION, BEAMFORMING, AND ASR

In this section, we first describe the proposed architecture for end-to-end dereverberation, beamforming (denoising and separation), and ASR. And the formulation of different beamformer variants supported in the proposed framework is given. We then introduce the techniques applied to solve the numerical instability issue. Later, the frequency permutation phenomenon and our solution are discussed.

2.1. Model architecture with advanced frontend

Our proposed end-to-end architecture is shown in Fig. 1, which is comprised of two main modules: the frontend (speech enhancement) and the backend (ASR). Here, speech enhancement includes dereverberation, denoising and source separation. In our previous

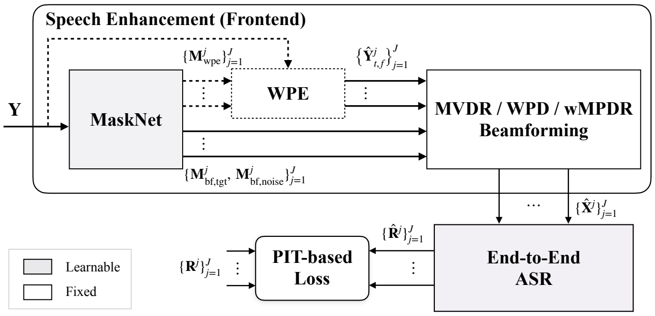
Speech Enhancement (Frontend)
Y
MaskNet
$\{\mathbf{M}^j_{\text{wpe}}\}_{j=1}^J$
WPE
$\{\hat{\mathbf{Y}}^j_{t,f}\}_{j=1}^J$
MVDR / WPD / wMPDR Beamforming
$\{\mathbf{M}^j_{\text{bf,tgt}}, \mathbf{M}^j_{\text{bf,noise}}\}_{j=1}^J$
$\{\hat{\mathbf{X}}^j\}_{j=1}^J$
$\{\hat{\mathbf{R}}^j\}_{j=1}^J$
$\{\mathbf{R}^j\}_{j=1}^J$
PIT-based Loss
End-to-End ASR
Learnable
Fixed

Fig. 1: Proposed new architecture for end-to-end training of the frontend and ASR backend.

article [22], we adopted a weighted power minimization distortionless response (WPD) convolutional beamformer [25] as a unified frontend, while the recent study [26] showed that a WPD can be factorized into a WPE dereverberation filter and a weighted minimum power distortionless response (wMPDR) [26] beamformer without loss of optimality when they are jointly optimized. Therefore, in this article, we adopt the factorized form as a simpler alternative[1]. That is, the frontend is composed of a single mask estimator (`MaskNet`), a DNN-WPE [27] dereverberation module, and a beamformer module. In addition, we mainly support two alternative beamformer types, respectively, based on 1) minimum variance distortionless response (MVDR) [28] and 2) wMPDR. While MVDR is a widely used state-of-the-art beamformer, wMPDR is shown to perform optimal processing jointly with WPE [26]. The ASR backend is a joint connectionist temporal classification (CTC) / attention-based encoder-decoder [13] model for recognizing the separated single-channel speech. Compared to those in our previous work [22], the proposed architecture can support different beamformer variants in a single framework, by using a single mask estimator for WPE / beamforming and applying single-source WPE for processing speech of different sources.

Below we give a detailed description of the proposed system. Consider a multichannel input speech signal composed of J speakers, $\mathbf{Y}_{t,f} = \{Y_{t,f,c}\}_{c=1}^{C} \in \mathbb{C}^C$, it can be described as follows in the short-time Fourier transform (STFT) domain:

$$\mathbf{Y}_{t,f} = \sum_{j=1}^{J} \mathbf{X}^j_{t,f} + \mathbf{N}_{t,f} = \sum_{j=1}^{J} \mathbf{X}^{(\text{d}),j}_{t,f} + \mathbf{X}^{(\text{r}),j}_{t,f} + \mathbf{N}_{t,f} \tag{1}$$

$$\mathbf{X}^{(\text{d}),j}_{t,f} = \sum_{\tau=0}^{\Delta-1} \mathbf{a}^j_{\tau,f} s^j_{t-\tau,f} \approx \mathbf{v}^j_f s^j_{t,f}\,, \tag{2}$$

$$\mathbf{X}^{(\text{r}),j}_{t,f} = \sum_{\tau=\Delta}^{L_a} \mathbf{a}^j_{\tau,f} s^j_{t-\tau,f}\,, \tag{3}$$

where $C > 1$ denotes the number of microphones. $t \in \{1, \ldots, T\}$ and $f \in \{1, \ldots, F\}$ represent the indices of time and frequency bins. $\mathbf{N}$ denotes noise. $\mathbf{X}^j$ denotes the reverberant signal, which can be decomposed into an "early" part $\mathbf{X}^{(\text{d}),j}$ and a "late" part $\mathbf{X}^{(\text{r}),j}$. $\mathbf{X}^{(\text{d}),j}$ contains the direct path and early reflection of the j-th speaker, while $\mathbf{X}^{(\text{r}),j}$ denotes the late reverberation. $\mathbf{a}^j_{\tau,f}$ is the acoustic transfer function with length L_a. Δ denotes the starting frame for the "late" part. s^j is the j-th source signal. $\mathbf{v}^j_f = \{v^j_{f,c}\}_{c=1}^{C} \in \mathbb{C}^C$ is the steering vector (SV). The input signal is first processed by the frontend module for dereverberation and separation. First, the WPE submodule performs dereverberation separately for each source j directly on the mixture $\mathbf{Y}$ in Eq.(1):

$$\{\mathbf{M}^j_{\text{wpe}}\}_{j=1}^J,\ \{\mathbf{M}^j_{\text{bf,tgt}}\}_{j=1}^J,\ \{\mathbf{M}^j_{\text{bf,noise}}\}_{j=1}^J = \mathrm{MaskNet}(\mathbf{Y})\,, \tag{4}$$

$$\lambda^j_{t,f} = \frac{1}{C} \sum_{c=1}^{C} \frac{M^j_{\text{wpe},t,f,c}}{\frac{1}{T}\sum_{\tau=1}^{T} M^j_{\text{wpe},\tau,f,c}} |Y_{t,f,c}|^2 \quad \in \mathbb{R}\,, \tag{5}$$

$$\hat{\mathbf{Y}}^j = \mathbf{Y}^j_{\text{wpe}} = \mathrm{WPE}(\mathbf{Y}, \boldsymbol{\lambda}^j) \qquad \in \mathbb{C}^{T\times F\times C}\,. \tag{6}$$

Here, $\mathbf{M}^j_{\text{wpe}} = \{M^j_{\text{wpe},t,f,c}\}_{t,f,c}$ denotes the estimated dereverberation mask, $\mathbf{M}^j_{\text{bf,tgt}}$ and $\mathbf{M}^j_{\text{bf,noise}}$ denote the estimated speech mask and distortion mask for the j-th speaker, respectively. $\boldsymbol{\lambda}^j = \{\lambda_{t,f}\}^j_{t,f}$ is the estimated time-varying power of the speech signal. $\mathrm{WPE}(\cdot)$ represents the dereverberation filter computation based on the WPE algorithm described in [29], and the detailed formulas are omitted here for simplicity. The signal $\hat{\mathbf{Y}}$ is then denoised and separated by the neural beamformer. Within the scope of this paper, although different beamformers are designed for different objectives with a linear constraint, their solutions can be uniformly written as:

$$\boldsymbol{\Phi}^j_{\alpha,f} = \frac{\sum_{t=1}^{T} \overline{M}^j_{t,f} \hat{\mathbf{Y}}^j_{t,f} \big(\hat{\mathbf{Y}}^j_{t,f}\big)^{\mathsf{H}}}{\sum_{t=1}^{T} \overline{M}^j_{t,f}} \qquad \in \mathbb{C}^{C\times C}\,, \tag{7}$$

$$\mathbf{w}^j_f = \frac{\big(\boldsymbol{\Phi}^j_{\text{N},f}\big)^{-1} \boldsymbol{\Phi}^j_{\text{S},f}}{\text{Trace}\Big[\big(\boldsymbol{\Phi}^j_{\text{N},f}\big)^{-1} \boldsymbol{\Phi}^j_{\text{S},f}\Big]} \mathbf{u}\,, \qquad \text{[w/o SV]} \tag{8}$$

$$= \frac{\big(\boldsymbol{\Phi}^j_{\text{N},f}\big)^{-1} \mathbf{v}^j_f}{\big(\mathbf{v}^j_f\big)^{\mathsf{H}} \big(\boldsymbol{\Phi}^j_{\text{N},f}\big)^{-1} \mathbf{v}^j_f} \big(v^j_{f,q}\big)^*\,, \qquad \text{[w/ SV]} \tag{9}$$

$$\hat{X}^j_{t,f} = \big(\mathbf{w}^j_f\big)^{\mathsf{H}} \hat{\mathbf{Y}}_{t,f} \qquad \in \mathbb{C}\,, \tag{10}$$

where $\overline{M}^j_{t,f} = \frac{1}{C}\sum_{c=1}^{C} M^j_{t,f,c}$ is a channel-averaged mask, where $M^j_{t,f,c} \in \{M^j_{\text{bf,tgt},t,f}, M^j_{\text{bf,noise},t,f}\}$, and $\boldsymbol{\Phi}^j_{\alpha,f}$ is a covariance matrix with a subscript $\alpha \in \{\text{N}, \text{S}, \text{noise}\}$. We set $\overline{M}^j_{t,f} = \overline{M}^j_{\text{bf,noise},t,f}$ for $\boldsymbol{\Phi}^j_{\text{noise},f}$ and $\overline{M}^j_{t,f} = \overline{M}^j_{\text{bf,tgt},t,f}$ for $\boldsymbol{\Phi}^j_{\text{S},f}$. Similarly, we set $\overline{M}^j_{t,f} = \overline{M}^j_{\text{bf,noise},t,f}$ and $\overline{M}^j_{t,f} = 1/\lambda^j_{t,f}$, respectively, for $\boldsymbol{\Phi}^j_{\text{N},f}$ of MVDR and wMPDR. $(\cdot)^*$ and $(\cdot)^{\mathsf{H}}$ denote conjugate and conjugate transpose, respectively. $\mathbf{w}^j_f$ is the beamforming filter for the j-th speaker, which can be calculated with either Eq. (8) [w/o SV] or Eq. (9) [w/ SV]. While Eq. (8) has been widely used for the E2E training of neural beamformers [17, 21], Eq. (9) is a standard equation for distortionless beamformers. $\mathbf{u}$ is a vector denoting the reference channel, which can be estimated by the attention mechanism [30], or based on the average estimated a posteriori SNR [15], or manually set as a one-hot vector. The subscript q denotes the reference channel index. $\hat{X}^j_{t,f}$ is the beamformed signal. $\mathbf{v}^j_f$ can be calculated through the eigendecomposition [31]:

$$\mathbf{v}^j_f = \boldsymbol{\Phi}^j_{\text{noise},f}\, \mathrm{MaxEigVec}\Big[\big(\boldsymbol{\Phi}^j_{\text{noise},f}\big)^{-1} \boldsymbol{\Phi}^j_{\text{S},f}\Big] \quad \in \mathbb{C}^C\,, \tag{11}$$

where $\mathrm{MaxEigVec}[\cdot]$ calculates the eigenvector corresponding to the maximum eigenvalue. Due to the lack of complex eigendecomposition support in PyTorch at the time of writing, we replace it with the power iteration method [32], which can be easily implemented for back-propagation, with a slight loss of precision.

It is worth noting that in the sense of end-to-end training, the MVDR and wMPDR beamformers are potentially equivalent. By substituting the $\boldsymbol{\Phi}^j_{\text{N},f}$ for wMPDR defined above into Eq. (8) or Eq. (9), we can find that the average operation in the denominator of Eq. (5) is canceled. Thus the derived wMPDR filter only depends on the (inversed) mask predicted by the neural network, which is very

[1] The experimental result on WPD is also given in Table 2 for comparison.

similar to the MVDR formulation. So it is hard to tell which beamformer is actually learned by the network via end-to-end training.

Finally, the separated stream $\hat{\mathbf{X}}^j = \{\hat{X}^j_{t,f}\}_{t,f}$ of each speaker j is fed into the ASR backend for recognition. First, the log Mel-filterbank coefficients $\mathbf{O}^j = \{\mathbf{o}^j_1, \ldots, \mathbf{o}^j_T\}$ with global mean and variance normalization (GMVN-LMF$(\cdot)$) is extracted from $\hat{\mathbf{X}}^j$, which is then transformed by the encoder into a high-level representation $\mathbf{H}^j = \{\mathbf{h}^j_1, \ldots, \mathbf{h}^j_L\}$ $(L \leq T)$ with subsampling. In order to solve the label ambiguity problem with multiple speakers $(J > 1)$, the permutation invariant training (PIT) technique [33] is applied in the CTC module to determine the order of the label sequences. With the best permutation derived in CTC, the representation $\mathbf{H}^j$ is processed by the attention-based decoder to generate the output token sequences $\hat{\mathbf{R}}^j = \{\hat{R}^j_1, \ldots, \hat{R}^j_N\}$ with length N, while $\mathbf{R}^j$ in Fig. 1 is the corresponding reference label. The speech recognition process for each speaker j is formulated as follows:

$$\mathbf{O}^j = \text{GMVN-LMF}(|\hat{\mathbf{X}}^j|)\,, \tag{12}$$

$$\mathbf{H}^j = \text{Encoder}(\mathbf{O}^j)\,, \tag{13}$$

$$\hat{R}^j_n \sim \text{Attention-Decoder}(\mathbf{H}^j, \hat{R}^j_{n-1})\,, \tag{14}$$

where $\hat{R}^j_n$ is the output token at the n-th decoding step.

Note that the entire system is optimized with sorely the ASR loss, which is a combination of the attention and CTC losses.

2.2. Attacking the numerical instability issue

The numerical instability issue has been a well-known problem in the beamformer [34], especially when optimized in an end-to-end manner. The numerical problem generally originates from the complex operations in the WPE and beamforming formulas, such as the complex matrix inverse, leading to poor performance in certain frequency bins sparsely populated. Such behaviors are particularly undesirable in the joint training with ASR, as they can easily result in not-a-number (NaN) gradients that fail to backpropagate correctly and even prevent the model from converging properly [22], thus badly impacting the overall model performance. In order to mitigate this problem, we propose four approaches to improve the stability of both WPE and beamforming submodules:

(1) Diagonal loading In order to stabilize the matrix inverse operation in WPE and beamforming in Eqs. (6), (8), (9) and (11), particularly at its backward pass, we introduce a diagonal loading [34] term as a perturbation to the complex matrix $\mathbf{\Phi}$ before inversion:

$$\mathbf{\Phi}' = \mathbf{\Phi} + \varepsilon\,\text{Trace}(\mathbf{\Phi})\mathbf{I}\,, \tag{15}$$

where $\mathbf{I}$ is the identity matrix, and ε is a tiny constant. For better stabilization, $\text{Trace}(\mathbf{\Phi})$ is used to make the term adaptive to signal level, and ε was set at a relatively large value for WPE in our experiments, as described in Section 3.1.

(2) Mask flooring When optimizing masks with an implicit criterion, i.e. the ASR loss, we observed that the mask estimator learned to predict sparse or spiky masks. This means, the mask estimator sets only the most relevant time-frequency bins to one, and the remaining ones to zero. It can then result in a singular covariance matrix in some frequency bins, making the WPE / beamforming process unstable. To avoid the spiky masks, we propose a mask flooring operation to introduce some regularization to the masks in Eq. (4):

$$\hat{M}_{t,f} = \text{Maximum}\{M_{t,f},\, \xi\}\,, \tag{16}$$

where $\hat{M}_{t,f}$ denotes the floored mask value, $M_{t,f} \in \{M_{\text{wpe},t,f,c}, \overline{M}_{\text{bf,tgt},t,f}, \overline{M}_{\text{bf,noise},t,f}\}$, and ξ is a constant flooring factor. The idea of the flooring is, that enough values have to be nonzero to reduce the effect of the flooring value. So the mask estimator is prevented from predicting sparse or spiky masks.

(3) More stable complex matrix operations Due to the lack of complex support in PyTorch, the alternative method in Section 4.3 in [35] was used in our previous work [22], which tries to find a factor to construct an invertible real matrix and maps the complex inversion to some real matrix operations. But it sometimes fails due to the poor estimate of the factor that results in a singular matrix. In this paper, a more stable matrix inverse formula [36] is implemented, which converts the problem of complex matrix inverse $\mathbf{\Phi}^{-1} = (\mathbf{A} + i\mathbf{B})^{-1} \in \mathbb{C}^{m \times m}$ into the inverse of a $2m \times 2m$ real matrix:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ -\mathbf{B} & \mathbf{A} \end{bmatrix}^{-1} = \begin{bmatrix} \mathcal{R}\{\mathbf{\Phi}^{-1}\} & \mathcal{I}\{\mathbf{\Phi}^{-1}\} \\ -\mathcal{I}\{\mathbf{\Phi}^{-1}\} & \mathcal{R}\{\mathbf{\Phi}^{-1}\} \end{bmatrix}\,, \tag{17}$$

where $\mathcal{R}\{\cdot\}$ and $\mathcal{I}\{\cdot\}$ denotes the real and imaginary parts of a complex matrix. Furthermore, we replace the inverse and the subsequent multiplication operations in Eqs. (6), (8) and (9) with a solve operation, which directly computes the solution $\mathbf{x}$ to a linear matrix equation $\mathbf{\Phi}\mathbf{x} = \mathbf{v}$, where $\mathbf{x}$ and $\mathbf{v}$ are m-dimensional vectors. It further improves the numerical accuracy and stability.[2]

(4) Double precision In terms of the implementation, while the end-to-end systems normally operate with the single-precision data / parameters, we find it beneficial to use the double precision for complex operations in the frontend module. It can reduce the error caused by complex operations, such as the inverse of close-to-singular matrices. Thus the stability of matrix inverse related operations can also be improved. Similar effects are also reported in [37], which proposes to jointly optimize the WPE and acoustic models.

With the above proposed techniques, we are now able to optimize the convolutional beamformer and ASR jointly, without the need of pretraining as in [22].

2.3. VAD-like mask for WPE and beamforming

During the end-to-end optimization of the frontend and backend, we often observed that beamformer outputs corresponding to different speakers are permuted with each other at certain frequencies. This is known as the frequency permutation problem [38]. It is probably caused by the fact that beamforming filters are estimated independently at each frequency bin with the predicted time-frequency (T-F) masks, and that the log Mel-filterbank features used for evaluating the ASR loss are obtained by averaging frequency bins with a triangle window, thus largely reducing the influence of the permutation errors on the loss. This, however, is not optimal for speech enhancement in the frontend. To solve this problem, instead of using T-F masks in Eq. (4), we propose to use the voice activity detection (VAD) like masks [19, 24], which share the same (soft) value over the frequency axis. This mask will be shown effective to mitigate the frequency permutation problem in our experiments..

3. EXPERIMENTS

3.1. Experimental setup

In this section, we evaluate our proposed framework on the artificially generated spatialized wsj1-2mix dataset [21], which contains anechoic and reverberant versions of multichannel two-speaker speech mixtures. We trained our models on a multi-condition training subset, including both reverberant and anechoic training samples in the spatialized wsj1-2mix (98.5 hr ×2), and WSJ train_si284 single-speaker clean data (81.5 hr, only for training ASR). Since the proposed framework jointly optimizes the frontend and backend with the ASR loss, no parallel clean data is required for training. The development and evaluation subsets only contain reverberant samples from the spatialized wsj1-2mix, with the duration of 1.3 hr and

[2] The new implementations `inverse2` and `solve` are now available at `https://github.com/kamo-naoyuki/pytorch_complex`.

Table 1: Evaluation of the proposed techniques with the WPE + MVDR + ASR model of different architectures on the spatialized reverberant wsj1-2mix evaluation set. The number of filter taps K and channels C are set to 5 and 2 for evaluation (same as training), respectively.[3]

Architecture	WER (%)	PESQ	STOI	SDR (dB)
Original mixture	-	1.20	0.65	-1.45
Arch in [22]	21.88	1.12	0.62	1.23
+ (1) Diagonal loading	15.51	1.32	0.74	**3.20**
+ (2) Mask flooring	20.13	1.24	0.71	1.14
+ (3) Stable complex op.	15.70	1.31	0.74	3.05
+ (4) Double precision	18.06	1.27	0.73	1.99
+ Tech (1)–(4)	15.18	1.31	0.74	2.85
Proposed arch + Tech (1)–(4)	**15.01**	1.31	0.74	2.81

0.8 hr respectively. For feature extraction, the STFT is performed on the 16-kHz input speech with a 25-ms Hann window and a 10-ms frame shift, and the 257-dimensional spectral feature is extracted. For the ASR backend, 80-dimensional log Mel-filterbank features are extracted for each separated spectrum.

All our proposed models are implemented based on the ESPnet framework. The mask estimation network in Fig. 1 is a 3-layer bidirectional long-short term memory (BLSTM) network with 600 cells in each direction, followed by $J \times 3$ output layers, where the number of speakers is $J = 2$. The number of iterations for performing WPE is 1. During training, the number of channels C and WPE filter taps K are fixed to 2 and 5, respectively. In the ASR backend, we followed the same configurations in [39]. We set ε in Eq. (15) to 10^{-3} and 10^{-8} for WPE and beamforming, respectively. The mask flooring factor ξ in Eq. (16) is set to 10^{-6} and 10^{-2} for WPE and beamforming, respectively. The number of iterations for estimating the steering vector using the power iteration is set to 2. The reference channel q in Eq.(8) is set to 1. The Noam optimizer with 25000 warmup steps and an initial learning rate of 1.0 was used for training.

3.2. Experimental results

We first evaluate the proposed techniques in Section 2.2 in both previously used [22] and the proposed architectures, as shown in Table 1. For speech recognition, we use the word error rate (WER) for evaluation. The speech enhancement (SE) performance is evaluated using three common metrics: signal-to-distortion ratio (SDR) [40], short-time objective intelligibility (STOI) [41] and perceptual evaluation of speech quality score (PESQ) [42]. And the clean source signal from WSJ is adopted as the reference signal. In Table 1, we can observe that all proposed techniques can bring significant performance improvement compared to the baseline architecture in [22]. And the combination of the four techniques can further achieve a better ASR result, with improved speech enhancement performance. This illustrates the effectiveness of the proposed approaches. The last row shows that with the proposed techniques, our proposed architecture in Fig.1 can also achieve comparable performance.

Table 2: Evaluation of different beamformer variants and mask types on the spatialized reverberant wsj1-2mix evaluation set. "w/ SV" and "w/o SV" in Eq. (8)–(9) denote with and without explicit use of the steering vector, respectively.

ID	Model (+ASR)	Formula	Mask	WER	PESQ	STOI	SDR
1	WSJ eval92 [4]	-	-	4.4	-	-	-
2	WPE+MVDR [22]	w/o SV	T-F	15.72	1.15	0.62	0.62
3	WPD [22]	w/o SV	T-F	13.97	1.33	0.68	0.38
4	MVDR	w/o SV	T-F	11.66	1.46	0.80	6.48
5	WPE+MVDR			9.50	1.56	0.83	7.73
6	WPE+wMPDR			9.44	1.63	0.82	8.49
7	WPE+MVDR	w/ SV	T-F	**9.02**	1.50	0.83	6.93
8	WPE+wMPDR			9.23	1.54	0.82	7.12
9	WPE+MVDR	w/o SV	VAD	9.45	1.95	0.86	**12.54**
10	WPE+wMPDR			10.26	1.97	0.86	12.20

We then evaluate the proposed architectures with different beamformer variants under different configurations of filter taps $K \in \{1, 3, 5, 7, 10\}$, while the number of channels C is fixed to 6, and only present the best performance of each model in Table 2 due to the limited space. We also present the best ASR results from [22] in rows 2 and 3 for comparison, and the SE performance are also evaluated. Comparing rows 2 & 5 and rows 3 & 6, we can observe that the proposed methods greatly improve the ASR and SE performances compared to the previous systems, which attributes to the proposed techniques for mitigating the numerical instability issue. From row 4 to row 5, the performance gain indicates the DNN-WPE submodule plays an important role in our proposed architecture. Comparing the second and third sections in Table 2, the MVDR and wMPDR beamformers show very similar results based on the formulas in either Eq. (8) [w/o SV] or Eq. (9) [w/ SV]. This also indicates the potential equivalence of these beamformers in the end-to-end training, as mentioned in Section 2.1. And the latter formula tends to yield better ASR results with end-to-end training. When comparing the second and the last sections in Table 2, we can find that the proposed VAD-like masks are beneficial for the SE performance, with obvious improvement on PESQ, STOI and SDR. This indicates that the VAD-like mask can effectively mitigate the frequency permutation problem, thus improving the SE performance. Since the evaluation set is generated based on the WSJ eval92 subset, the first row in Table 2 can be regarded as the topline for our system. And we can observe that the proposed models with different beamformer variants can all achieve very good ASR performance, with an only ~5% higher WER than the topline on WSJ.

4. CONCLUSIONS

In this paper, we propose a robust framework for end-to-end training of dereverberation, beamforming (denoising and separation), and speech recognition. Four techniques are proposed to regularize and stabilize the WPE / beamforming process in the frontend module, which are shown to effectively improve the numerical stability. Different beamformer variants and mask types are compared in our proposed framework. Our experiments on the spatialized wsj1-2mix corpus show that the proposed end-to-end system can achieve fairly good ASR results, with also decent speech enhancement performance in the reverberant multi-speaker condition, while only optimized with the ASR criterion. In our future work, we would like to investigate the end-to-end training in realistic and more challenging conditions.

5. ACKNOWLEDGEMENT

Wangyou Zhang and Yanmin Qian were supported by the China NSFC projects (No. 62071288 and U1736202). The work reported here was started at JSALT 2020 at JHU, with support from Microsoft, Amazon and Google. Experiments were carried out on the PI supercomputers at Shanghai Jiao Tong University.

6. REFERENCES

[1] Y. Liu and D. Wang, "Divide and conquer: A deep CASA approach to talker-independent monaural speaker separation,"

[3] More detailed results can be found at https://speechlab.sjtu.edu.cn/members/wangyou-zhang/icassp21-material.pdf.

IEEE/ACM Trans. ASLP., vol. 27, no. 12, pp. 2092–2102, 2019.

[2] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. IEEE ICASSP*, 2020, pp. 46–50.

[3] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *arXiv:2002.08933*, 2020.

[4] S. Karita *et al.*, "A comparative study on transformer vs RNN in speech applications," in *Proc. IEEE ASRU*, 2019, pp. 449–456.

[5] Z. Tüske *et al.*, "Single headed attention based sequence-to-sequence model for state-of-the-art results on switchboard-300," *arXiv:2001.07263*, 2020.

[6] D. S. Park *et al.*, "Improved noisy student training for automatic speech recognition," *arXiv:2005.09629*, 2020.

[7] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[8] Y. Qian, C. Weng, X. Chang, S. Wang, and D. Yu, "Past review, current progress, and challenges ahead on the cocktail party problem," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 40–63, 2018.

[9] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "BLSTM supported GEV beamformer front-end for the 3rd CHiME challenge," in *Proc. IEEE ASRU*, 2015, pp. 444–451.

[10] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *Proc. ISCA Interspeech*, 2016, pp. 545–549.

[11] S. Settle, J. Le Roux, T. Hori, S. Watanabe, and J. R. Hershey, "End-to-end multi-speaker speech recognition," in *Proc. IEEE ICASSP*, 2018, pp. 4819–4823.

[12] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *Proc. IEEE ICASSP*, 2018, pp. 686–690.

[13] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. IEEE ICASSP*, 2017, pp. 4835–4839.

[14] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. IEEE ICASSP*, Mar. 2016, pp. 196–200.

[15] H. Erdogan *et al.*, "Improved MVDR beamforming using single-channel mask prediction networks," in *Proc. ISCA Interspeech*, 2016, pp. 1981–1985.

[16] X. Xiao *et al.*, "Deep beamforming networks for multi-channel speech recognition," in *Proc. IEEE ICASSP*, 2016, pp. 5745–5749.

[17] T. Ochiai *et al.*, "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, 2017.

[18] J. Heymann *et al.*, "Beamnet: End-to-end training of a beamformer-supported multi-channel ASR system," in *Proc. IEEE ICASSP*, 2017, pp. 5325–5329.

[19] A. S. Subramanian *et al.*, "An investigation of end-to-end multichannel speech recognition for reverberant and mismatch conditions," *arXiv:1904.09049*, 2019.

[20] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 7, pp. 1717–1731, 2010.

[21] X. Chang, W. Zhang, Y. Qian, J. Le Roux, and S. Watanabe, "MIMO-Speech: End-to-end multi-channel multi-speaker speech recognition," in *Proc. IEEE ASRU*, 2019, pp. 237–244.

[22] W. Zhang *et al.*, "End-to-end far-field speech recognition with unified dereverberation and beamforming," in *Proc. ISCA Interspeech*, 2020, pp. 324–328.

[23] C. Y. Lim, C.-H. Chen, and W.-Y. Wu, "Numerical instability of calculating inverse of spatial covariance matrices," *Statistics & Probability Letters*, vol. 129, pp. 182–188, 2017.

[24] Y.-H. Tu *et al.*, "An iterative mask estimation approach to deep learning based multi-channel speech recognition," *Speech Communication*, vol. 106, pp. 31–43, 2019.

[25] T. Nakatani and K. Kinoshita, "A unified convolutional beamformer for simultaneous denoising and dereverberation," *IEEE Signal Processing Letters*, vol. 26, no. 6, pp. 903–907, 2019.

[26] C. Boeddeker *et al.*, "Jointly optimal dereverberation and beamforming," in *Proc. IEEE ICASSP*, 2020, pp. 216–220.

[27] K. Kinoshita *et al.*, "Neural network-based spectrum estimation for online WPE dereverberation," in *Proc. ISCA Interspeech*, 2017, pp. 384–388.

[28] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[29] L. Drude *et al.*, "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.

[30] T. Ochiai *et al.*, "Multichannel end-to-end speech recognition," in *Proc. ICML*, 2017, pp. 2632–2641.

[31] N. Ito, S. Araki, M. Delcroix, and T. Nakatani, "Probabilistic spatial dictionary based online adaptive beamforming for meeting recognition in noisy and reverberant environments," in *Proc. IEEE ICASSP*, 2017, pp. 681–685.

[32] R. Mises and H. Pollaczek-Geiringer, "Praktische verfahren der gleichungsauflösung." *ZAMM—Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 9, no. 2, pp. 152–164, 1929.

[33] D. Yu *et al.*, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. IEEE ICASSP*, 2017, pp. 241–245.

[34] S. Chakrabarty and E. A. Habets, "On the numerical instability of an LCMV beamformer for a uniform linear array," *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 272–276, 2015.

[35] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," *Technical Univ. Denmark, Tech. Rep*, vol. 3274, 2012.

[36] W. Smith and S. Erdman, "A note on the inversion of complex matrices," *IEEE Transactions on Automatic Control*, vol. 19, no. 1, pp. 64–64, 1974.

[37] J. Heymann *et al.*, "Joint optimization of neural network-based WPE dereverberation and acoustic model for robust online ASR," in *Proc. IEEE ICASSP*, 2019, pp. 6655–6659.

[38] M. Z. Ikram *et al.*, "A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation," in *Proc. IEEE ICASSP*, 2002, pp. 881–884.

[39] X. Chang, W. Zhang, Y. Qian, J. Le Roux, and S. Watanabe, "End-to-end multi-speaker speech recognition with transformer," in *Proc. IEEE ICASSP*, 2020, pp. 6129–6133.

[40] C. Févotte *et al.*, "BSS-EVAL Toolbox User Guide—Revision 2.0," IRISA, Tech. Rep. 1706, Apr. 2005.

[41] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Trans. ASLP.*, vol. 24, no. 11, pp. 2009–2022, 2016.

[42] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)—a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE ICASSP*, vol. 2, 2001, pp. 749–752.