End-to-End Dereverberation, Beamforming, and Speech Recognition in A Cocktail Party

Wangyou Zhang, *Student Member, IEEE,* Xuankai Chang, *Student Member, IEEE,* Christoph Boeddeker, *Student Member, IEEE,* Tomohiro Nakatani, *Fellow, IEEE,* Shinji Watanabe, *Senior Member, IEEE,* and Yanmin Qian, *Senior Member, IEEE*

***Abstract*—Far-field multi-speaker automatic speech recognition (ASR) has drawn increasing attention in recent years. Most existing methods feature a signal processing frontend and an ASR backend. In realistic scenarios, these modules are usually trained separately or progressively, which suffers from either inter-module mismatch or a complicated training process. In this paper, we propose an end-to-end multi-channel model that jointly optimizes the speech enhancement (including speech dereverberation, denoising, and separation) frontend and the ASR backend as a single system. To the best of our knowledge, this is the first work that proposes to optimize dereverberation, beamforming, and multi-speaker ASR in a fully end-to-end manner. The frontend module consists of a weighted prediction error (WPE) based submodule for dereverberation and a neural beamformer for denoising and speech separation. For the backend, we adopt a widely used end-to-end (E2E) ASR architecture. It is worth noting that the entire model is differentiable and can be optimized in a fully end-to-end manner using only the ASR criterion, without the need of parallel signal-level labels. We evaluate the proposed model on several multi-speaker benchmark datasets, and experimental results show that the fully E2E ASR model can achieve competitive performance on both noisy and reverberant conditions, with over 30% relative word error rate (WER) reduction over the single-channel baseline systems.**

***Index Terms*—End-to-end, dereverberation, beamforming, speech separation, multi-talker speech recognition**

I. Introduction

In recent years, with the rapid development of deep learning, much progress has been achieved in single-speaker automatic speech recognition (ASR), even with performance on a par with humans [1], [2]. However, there still remains a large performance gap between single-speaker and multi-speaker conditions [3], [4]. Recently, more and more researchers have drawn their interest in tackling the multi-speaker speech recognition problem in the so-called cocktail party scenario [5], where multiple sound sources coexist with the presence of noise and reverberation. It is much more challenging than in clean and anechoic conditions, and the ASR performance on the multi-speaker overlapped speech is still far from satisfactory.

Wangyou Zhang and Yanmin Qian are with the X-LANCE Lab, Department of Computer Science and Engineering and MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, P. R. China ({wyz-97, yanminqian}@sjtu.edu.cn).

Xuankai Chang and Shinji Watanabe are with the Language Technologies Institute at Carnegie Mellon University, Pittsburgh, PA, USA ({xuankaic, swatanab}@andrew.cmu.edu).

Christoph Boeddeker is with Paderborn University, Paderborn, Germany (boeddeker@nt.upb.de).

Tomohiro Nakatani is with the NTT Corporation, Kyoto 619-0237, Japan (tnak@ieee.org).

While existing multi-speaker ASR methods can be categorized into single-channel and multi-channel, we focus on the latter one in this paper because additional spatial information can be leveraged to boost the performance. One straightforward way is to directly extend the single-channel approaches [3], [6]–[8] by incorporating the spatial feature into the original architecture [9], [10]. Another widely adopted method is known as the neural beamformer [11], [12], which integrates deep learning based approaches into the conventional beamforming [13] module. The neural beamformer is often favored for its good compatibility with the downstream ASR task, as it explicitly constrains the distortion of the desired signal and thus enjoys better generalizability in unseen conditions. The neural beamformer and ASR models are usually trained with separate losses [14] or in a progressive manner [15] with warm-start.

More recently, there has been increasing interest in end-to-end (E2E) training of neural beamformer based frontend and ASR backend modules, which solely uses the final ASR loss to optimize the entire system. This type of training scheme can naturally work around the limitations of the aforementioned separate or progressive training schemes, where parallel clean reference signals are required in the training stage. However, existing research on E2E training of frontend and backend modules either only focuses on single-speaker scenarios [16]–[20] or mainly considers anechoic and noise-free conditions [21], [22]. Thus, it is still unclear whether the fully E2E training is feasible in more realistic environments such as the cocktail party scenario where background noise, reverberation, and interference speakers are present.

In this paper, we aim to fill this gap and provide more insights into the practical issues when applying E2E training in noisy and reverberant multi-speaker conditions. We extend the prior study on this problem and investigate the effectiveness of the proposed methods in various conditions. The main novelties of this paper are summarized below:

(1) This is the first work that proposes an E2E dereverberation, beamforming, and multi-speaker ASR model, which can be trained in a fully E2E manner, with only the ASR criterion.
(2) We analyze the numerical instability issue in the frontend, which often impedes the success of E2E training of the proposed model. We propose to apply several techniques to greatly improve the numerical stability and system performance.
(3) We investigate the frequency permutation problem under

the E2E framework, and propose a 1-D mask approach and a frequency permutation adjustment strategy to significantly mitigate this problem.

(4) We present an extensive evaluation of the proposed model on several multi-speaker benchmark datasets, including spatialized WSJ0-2mix [23], SMS-WSJ [24], and WHAMR! [25].

(5) We propose several strategies to facilitate E2E training: multi-condition training, channel sampling, and approximated truncated back-propagation through time.

(6) We compare three different training schemes of the proposed model, i.e., fully E2E training, multi-task learning, and independent training of different modules.

This paper is an extension of our previous work [26], [27], which proposes and improves the E2E model for joint dereverberation, beamforming, and speech recognition in the multi-speaker scenario. In this paper, we first summarize the above efforts in a unified model with a tight and consistent formulation. Then several new training strategies are proposed to facilitate E2E training of the proposed model. In addition, more detailed experimental results and analyses are given. We extend the evaluation in the previous work [26], [27] from noise-free conditions to more realistic scenarios with background noise, and conduct extensive experiments on several multi-speaker benchmark datasets. Furthermore, thanks to the flexibility of the proposed model that allows using different training schemes, we compare the proposed fully E2E training scheme with other commonly adopted training schemes. The results on several multi-speaker benchmark datasets show that our proposed model can be easily adapted to different scenarios with a competitive performance.

II. Signal model and problem definition

In this paper, we focus on the speech recognition problem in the multi-channel multi-speaker scenario. In noisy conditions, we assume the observed signal $\mathbf{y} \in \mathbb{R}^C$ is composed of speech signals $\mathbf{x}^j$ from J different speakers and the background noise $\mathbf{n}$. C is the number of channels, and the superscript j denotes the j-th speaker. In the short-time Fourier transform (STFT) domain, the signal model is written as:

$$\mathbf{Y}_{t,f} = \sum_{j=1}^{J} \mathbf{X}_{t,f}^{j} + \mathbf{N}_{t,f} = \sum_{j=1}^{J} \left(\mathbf{X}_{t,f}^{(\mathrm{d}),j} + \mathbf{X}_{t,f}^{(\mathrm{r}),j} \right) + \mathbf{N}_{t,f}\,, \quad (1)$$

where the subscripts t and f denote the indices of time frames and frequency bins, respectively. $\mathbf{Y} \in \mathbb{C}^{T\times F\times C}$, $\mathbf{X} \in \mathbb{C}^{T\times F\times C}$, and $\mathbf{N} \in \mathbb{C}^{T\times F\times C}$ represent the spectrum of the observed signal, speech signal, and background noise, respectively. In reverberant conditions, the speech component $\mathbf{X}_{t,f}^{j} \in \mathbb{C}^C$ is further decomposed into an "early" part $\mathbf{X}_{t,f}^{(\mathrm{d}),j}$ and a "late" part $\mathbf{X}_{t,f}^{(\mathrm{r}),j}$, as shown in Eq. (1). The "early" signal $\mathbf{X}_{t,f}^{(\mathrm{d}),j}$ includes the direct path signal and early-arriving reflection of the j-th speaker, and the "late" signal $\mathbf{X}_{t,f}^{(\mathrm{r}),j}$ denotes the late reverberation. Assume the room impulse response (RIR) is longer than the STFT analysis window, the "early" and "late" signals are often defined as follows [28]:

$$\mathbf{X}_{t,f}^{(\mathrm{d}),j} = \sum_{\tau=0}^{\Delta-1} \mathbf{a}_{\tau,f}^{j} S_{t-\tau,f}^{j} \approx \mathbf{v}_{f}^{j} S_{t,f}^{j} \qquad \in \mathbb{C}^C\,, \quad (2)$$

$$\mathbf{X}_{t,f}^{(\mathrm{r}),j} = \sum_{\tau=\Delta}^{L_a} \mathbf{a}_{\tau,f}^{j} S_{t-\tau,f}^{j} \qquad \in \mathbb{C}^C\,, \quad (3)$$

where $\mathbf{a}_{f}^{j} \in \mathbb{C}^{L_a\times C}$ is the convolutional acoustic transfer function (ATF) from the j-th speaker to all microphones, and the subscript τ denotes taking the τ-th frame from each ATF whose total length is L_a. $\Delta > 0$ is the frame index in ATF from which on it is regarded to contribute to the late reverberation. $S_{t,f}^{j} \in \mathbb{C}$ denotes the clean source signal of the j-th speaker. In Eq. (2), the "early" signal is often simplified as the product of the source signal $S_{t,f}^{j}$ and the corresponding steering vector $\mathbf{v}_{f}^{j} \in \mathbb{C}^C$, based on the multiplicative transfer function approximation [29]. In the following discussion, we normalize the steering vector w.r.t. the reference channel, and refer to it as the relative transfer function (RTF).

Given the above signal model, the goal of *multi-talker dereverberation and separation* is to estimate the "early" signal $\mathbf{X}_{t,f}^{(\mathrm{d}),j}$ of each speaker j, while eliminating the late reverberation $\mathbf{X}_{t,f}^{(\mathrm{r}),j}$, interference speakers $\sum_{i\neq j} \mathbf{X}_{t,f}^{i}$, and background noise $\mathbf{N}_{t,f}$. Finally, the goal of *multi-talker speech recognition* is to generate the transcript corresponding to each speaker from the frontend output.

III. E2E Dereverberation, Beamforming, and multi-talker ASR

In this section, we first introduce the proposed differentiable E2E multi-channel model for joint speech enhancement (including speech dereverberation, denoising, and separation) and multi-speaker speech recognition. Then, we analyze the well-known numerical instability issue in the frontend processing and propose several techniques to alleviate this issue. Moreover, we revisit the frequency permutation problem under the E2E framework and propose a 1-D voice activity detection (VAD)-like mask as well as a frequency permutation adjustment strategy as a remedy. Moreover, we propose several strategies to facilitate E2E training. Finally, the relationship between different training schemes is discussed.

A. The proposed E2E multi-channel multi-talker model

Fig. 1 shows the overview of the proposed model. It is mainly composed of two cascaded modules: (1) the speech enhancement frontend for dereverberation, denoising, and speech separation; (2) the speech recognition backend for multi-talker ASR. The multi-channel input signal $\mathbf{Y}$ is first processed by the frontend module to generate separated signals $\{\hat{\mathbf{X}}^{j}\}_{j=1}^{J}$ for all speakers $j \in \{1, 2, \cdots, J\}$. Each separated stream is then fed into the multi-talker ASR backend separately to obtain the respective transcript. The feature extraction block bridges the two modules and makes the entire model fully differentiable, which enables E2E training of the whole system.

B. Speech enhancement frontend

The speech enhancement (SE) frontend consists of two main submodules: dereverberation and separation. Both submodules are based on the conventional signal processing approaches [13], [28], [30], which have been shown to yield low-distortion outputs [31] that are beneficial to both perceptual listening quality [32], [33] and the downstream speech recognition task [11], [12], [33]–[37]. More specifically, the

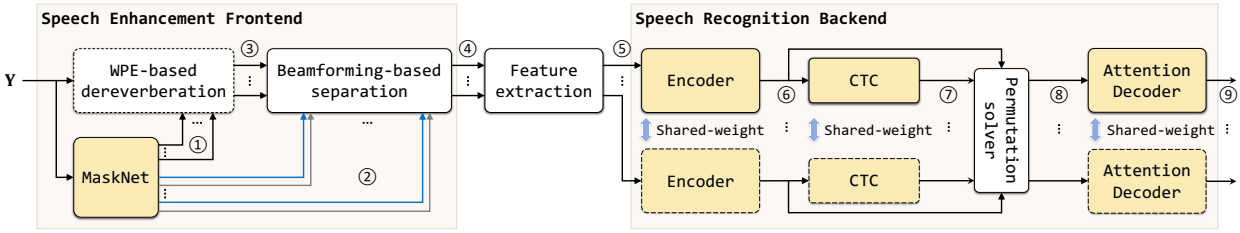


Fig. 1: Proposed E2E multi-channel model for joint dereverberation, beamforming, and multi-talker speech recognition. Learnable modules are depicted as ▭, while non-learnable modules are depicted as ▭. The annotated variables in the figure are: ① estimated masks $\{\mathbf{M}^j_{\text{wpe}}\}^J_{j=1}$ for WPE. ② estimated target speech masks $\{\mathbf{M}^j_{\text{bf-S}}\}^J_{j=1}$ and noise masks $\{\mathbf{M}^j_{\text{bf-N}}\}^J_{j=1}$ for beamforming. ③ dereverberated signals $\{\hat{\mathbf{Y}}^{(\text{d}),j}\}^J_{j=1}$ with respect to each speaker j. ④ separated signals $\{\hat{\mathbf{X}}^j\}^J_{j=1}$ after beamforming. ⑤ extracted features $\{\mathbf{O}^j\}^J_{j=1}$ for speech recognition. ⑥ encoder representations $\{\mathbf{H}^j\}^J_{j=1}$. ⑦ CTC losses$\{\mathcal{L}^j_{\text{ctc}}\}^J_{j=1}$ for all possible permutations of the separated streams and labels. ⑧ encoder representations $\{\hat{\mathbf{H}}^j\}^J_{j=1}$ rearranged according to the best permutation obtained from the permutation solver. ⑨ attention-decoder losses $\{\mathcal{L}^j_{\text{att}}\}^J_{j=1}$.

dereverberation submodule is based on the weighted prediction error (WPE) algorithm [30], and the separation submodule is based on the beamforming technique [11]–[13], including minimum variance distortionless response (MVDR) [38], minimum power distortionless response (MPDR) [39] beamformer and the recently proposed convolutional beamformers (wMPDR and WPD) [40]–[44]. Both submodules rely on a neural network, `MaskNet`, to estimate masks for calculating the required statistics, e.g., the correlation matrix and vector for WPE and the cross-channel power spectral density (PSD) matrices for beamforming. The former is previously known as DNN-WPE [45] in the context of single-speaker speech enhancement, and the latter is usually called a neural beamformer [11], [12]. For the cascade order of the two submodules, we opt to perform WPE-based dereverberation before beamforming, which has proven to be effective in various prior works [41], [46].

The detailed speech enhancement procedure (left part in Fig. 1) is described as follows. The input signal $\mathbf{Y}$ is first fed into `MaskNet` to obtain the masks for frontend processing[1]:

$$\{\mathbf{M}^j_{\text{wpe}}, \mathbf{M}^j_{\text{bf-S}}, \mathbf{M}^j_{\text{bf-N}}\}^J_{j=1} = \texttt{MaskNet}(\mathbf{Y}), \tag{4}$$

where $\mathbf{M}^j_{\text{wpe}} \in \mathbb{R}^{T\times F\times C}$ denotes the corresponding mask of the j-th speaker for WPE. $\mathbf{M}^j_{\text{bf-S}} \in \mathbb{R}^{T\times F\times C}$ and $\mathbf{M}^j_{\text{bf-N}} \in \mathbb{R}^{T\times F\times C}$ denote the target speech and noise masks w.r.t. the j-th speaker respectively for beamforming. Note that all masks used in this paper are magnitude-only masks, and the investigation of complex-valued masks is left for future work. F is the total number of frequency bins. It is worth noting that in the implementation of `MaskNet`, the masks are estimated for each input channel independently, which naturally allows processing a varying number of input channels with different array geometries. After mask estimation, the single-target WPE filter $\hat{\tilde{\mathbf{W}}}^j_f \in \mathbb{C}^{CK\times C}$ is estimated separately for each speaker j following Eqs. (6)–(8) in [47], while the time-varying variance for each speaker j is replaced with

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

where subscripts t, f, c denote taking the element of the t-th frame, f-th frequency bin, and c-th channel in a variable. The summation in the denominator normalizes the WPE mask along the time frame dimension. Note that different from the iterative process usually used in conventional WPE, the mask-based DNN-WPE can be applied with a single pass given the accurate mask estimation from `MaskNet`. The dereverberated signal $\hat{\mathbf{Y}}^{(\text{d}),j}_{t,f}$ for each speaker j is then obtained:

$$\hat{\mathbf{Y}}^{(\text{d}),j}_{t,f} = \mathbf{Y}_{t,f} - \left(\hat{\tilde{\mathbf{W}}}^j_f\right)^{\mathsf{H}} \tilde{\mathbf{Y}}_{t-\Delta,f} \quad \in \mathbb{C}^C, \tag{6}$$

where $\tilde{(\cdot)}$ denotes the stacked representations of a variable, e.g., $\tilde{\mathbf{Y}}_{t-\Delta,f} = \left[\mathbf{Y}^{\mathsf{T}}_{t-\Delta,f}, \mathbf{Y}^{\mathsf{T}}_{t-(\Delta+1),f}, \cdots, \mathbf{Y}^{\mathsf{T}}_{t-(K+\Delta-1),f}\right]^{\mathsf{T}} \in \mathbb{C}^{CK}$.

Meanwhile, the beamforming filter of various beamformer types can be also estimated for each speaker j based on the following unified formulas:

$$\mathbf{w}^{\mathcal{X},j}_f = \begin{cases} \dfrac{\left(\mathbf{\Phi}^{\mathcal{X},j}_{1,f}\right)^{-1}\mathbf{\Phi}^{\mathcal{X},j}_{2,f}}{\text{Trace}\left[\left(\mathbf{\Phi}^{\mathcal{X},j}_{1,f}\right)^{-1}\mathbf{\Phi}^{\mathcal{X},j}_{2,f}\right]}\mathbf{u}^{\mathcal{X}}, & \text{[w/o RTF]} \quad (7) \\ \dfrac{\left(\mathbf{\Phi}^{\mathcal{X},j}_{1,f}\right)^{-1}\mathbf{v}^{\mathcal{X},j}_f}{\left(\mathbf{v}^{\mathcal{X},j}_f\right)^{\mathsf{H}}\left(\mathbf{\Phi}^{\mathcal{X},j}_{1,f}\right)^{-1}\mathbf{v}^{\mathcal{X},j}_f}, & \text{[w/ RTF]} \quad (8) \end{cases}$$

where $\mathbf{\Phi}^{\mathcal{X},j}_{1,f}$ and $\mathbf{\Phi}^{\mathcal{X},j}_{2,f}$ are different covariance matrices defined by the specific beamformer type $\mathcal{X} \in$ {MVDR, MPDR, wMPDR, WPD[2]}. $(\cdot)^*$ denotes complex conjugate. $\mathbf{u}^{\mathcal{X}}$ is a one-hot reference channel selection vector. $\mathbf{v}^{\mathcal{X},j}_f$ is the RTF vector. The beamformer filter $\mathbf{w}^{\mathcal{X},j}_f$ for a specific beamformer type $\mathcal{X}$ can be formulated either dependent on the RTF [Eq. (8)] or based on the reference channel selection [Eq. (7)]. The definition of the beamformer-specific variables in Eqs. (7)–(8) is summarized in Table I.

More specifically, $\mathbf{\Phi}^j_{\text{S},f} \in \mathbb{C}^{C\times C}$ and $\mathbf{\Phi}^j_{\text{N},f} \in \mathbb{C}^{C\times C}$ in Table I are the speech and noise PSD matrices of the j-th source,

[1]Another possible design is using two separate DNNs for WPE and beamforming mask estimation, respectively, which has been discussed in our previous work [27]. Here, we followed the single `MaskNet` design to simplify the discussion.

[2]Note that the WPE submodule in Fig. 1 is unused for the WPD beamformer, as it is implicitly included inside the beamformer design.

TABLE I: Definition of variables in Eqs.(7)–(19) for different beamformer types.

Beamformer Type $\mathcal{X}$	$\mathbf{\Phi}_{1,f}^{\mathcal{X},j}$	$\mathbf{\Phi}_{2,f}^{\mathcal{X},j}$	$\mathbf{u}^{\mathcal{X}}$	$\mathbf{v}_f^{\mathcal{X},j}$	$\hat{\mathbf{Y}}_{t,f}^{\mathcal{X},j}$
MVDR MPDR wMPDR	$\mathbf{\Phi}_{\mathrm{N},f}^{j}$: Eq. (10) $\mathbf{\Phi}_{\mathrm{O},f} = (1/T)\sum_t [\mathbf{Y}_{t,f}\mathbf{Y}_{t,f}^{\mathsf{H}}]$ $\mathbf{\Phi}_{\mathrm{d},f}^{j}$: Eq. (11)	$\mathbf{\Phi}_{\mathrm{S},f}^{j}$: Eq. (9)	$\mathbf{u}$	$\mathbf{v}_f^j$: Eqs. (13)–(16)	$\hat{\mathbf{Y}}_{t,f}^{(\mathrm{d}),j}$: Eq. (6)
WPD	$\bar{\mathbf{R}}_f^j$: Eq. (17)	$\bar{\mathbf{\Phi}}_{\mathrm{S},f}^{j}$: Eq. (18)	$\bar{\mathbf{u}} = \left[\mathbf{u}^{\mathsf{T}}, \mathbf{0}, \cdots, \mathbf{0}\right]^{\mathsf{T}}$	$\bar{\mathbf{v}}_f^j = \left[(\mathbf{v}_f^j)^{\mathsf{T}}, \mathbf{0}, \cdots, \mathbf{0}\right]^{\mathsf{T}}$	$\bar{\mathbf{Y}}_{t,f} = \left[\mathbf{Y}_{t,f}^{\mathsf{T}}, \tilde{\mathbf{Y}}_{t-\Delta,f}^{\mathsf{T}}\right]^{\mathsf{T}}$

respectively:

$$\mathbf{\Phi}_{\mathrm{S},f}^{j} = \frac{\sum_{t=1}^{T}\left(\sum_{c=1}^{C} M_{\text{bf-S},t,f,c}^{j}\right)\hat{\mathbf{Y}}_{t,f}^{(\mathrm{d}),j}\left(\hat{\mathbf{Y}}_{t,f}^{(\mathrm{d}),j}\right)^{\mathsf{H}}}{\sum_{t=1}^{T}\sum_{c=1}^{C} M_{\text{bf-S},t,f,c}^{j}}, \tag{9}$$

$$\mathbf{\Phi}_{\mathrm{N},f}^{j} = \frac{\sum_{t=1}^{T}\left(\sum_{c=1}^{C} M_{\text{bf-N},t,f,c}^{j}\right)\hat{\mathbf{Y}}_{t,f}^{(\mathrm{d}),j}\left(\hat{\mathbf{Y}}_{t,f}^{(\mathrm{d}),j}\right)^{\mathsf{H}}}{\sum_{t=1}^{T}\sum_{c=1}^{C} M_{\text{bf-N},t,f,c}^{j}}. \tag{10}$$

$\mathbf{\Phi}_{\mathrm{d},f}^{j}$ in the wMPDR beamformer is the power-normalized PSD matrix, which can be estimated based on Eqs. (5)–(6):

$$\mathbf{\Phi}_{\mathrm{d},f}^{j} = \frac{1}{T}\sum_{t=1}^{T}\frac{\hat{\mathbf{Y}}_{t,f}^{(\mathrm{d}),j}\left(\hat{\mathbf{Y}}_{t,f}^{(\mathrm{d}),j}\right)^{\mathsf{H}}}{\lambda_{t,f}^{j}} \quad \in \mathbb{C}^{C\times C}. \tag{11}$$

To estimate the RTF vector $\mathbf{v}_f^j$, we adopt the covariance whitening based RTF estimation approach [48]:

$$\hat{\mathbf{v}}_f^j = \mathbf{\Phi}_{\mathrm{N},f}^{j}\,\mathrm{MaxEigVec}\left[\left(\mathbf{\Phi}_{\mathrm{N},f}^{j}\right)^{-1}\mathbf{\Phi}_{\mathrm{S},f}^{j}\right] \quad \in \mathbb{C}^{C}, \tag{12}$$

where $\mathrm{MaxEigVec}[\cdot]$ calculates the principal eigenvector of a complex matrix. Based on our preliminary experiments [26], [27], we resort to the power iteration method [49], which has proven to have high convergence speed while providing accurate RTF estimation [50], for approximating the eigenvalue decomposition result. Letting $\mathbf{D}_f^j = \left(\mathbf{\Phi}_{\mathrm{N},f}^{j}\right)^{-1}\mathbf{\Phi}_{\mathrm{S},f}^{j}$, the detailed procedure can be formulated as follows:

$$\text{Step 1) initialize:} \quad \hat{\mathbf{v}}_f^j \leftarrow \mathbf{u}, \tag{13}$$

$$\text{Step 2) iterate for } p \text{ times:} \quad \hat{\mathbf{v}}_f^j \leftarrow \mathbf{D}_f^j\hat{\mathbf{v}}_f^j, \tag{14}$$

$$\text{Step 3) estimate ATF[3]:} \quad \hat{\mathbf{v}}_f^j \leftarrow \mathbf{\Phi}_{\mathrm{N},f}^{j}\hat{\mathbf{v}}_f^j, \tag{15}$$

$$\text{Step 4) calculate RTF:} \quad \hat{\mathbf{v}}_f^j \leftarrow \hat{\mathbf{v}}_f^j/\hat{v}_f^{(\mathrm{b}),j}, \tag{16}$$

where Steps 1), 2) correspond to the power iteration algorithm for approximating the $\mathrm{MaxEigVec}[\cdot]$ operation in Eq. (12). The RTF $\hat{\mathbf{v}}_f^j$ is initialized in Step 1) as a one-hot vector $\mathbf{u} \in \mathbb{R}^C$ with the b-th element being 1. p is the number of iterations.

For the weighted power minimization distortionless response (WPD) beamformer [26], [40], $\bar{\mathbf{R}}_f^j$ and $\bar{\mathbf{\Phi}}_{\mathrm{S},f}$ in Table I are respectively the stacked power-normalized PSD matrix and zero-padded speech PSD matrix:

$$\bar{\mathbf{R}}_f^j = \frac{1}{T}\sum_{t=1}^{T}\frac{\bar{\mathbf{Y}}_{t,f}\bar{\mathbf{Y}}_{t,f}^{\mathsf{H}}}{\lambda_{t,f}^{j}} \quad \in \mathbb{C}^{C(K+1)\times C(K+1)}, \tag{17}$$

$$\bar{\mathbf{\Phi}}_{\mathrm{S},f}^{j} = \begin{bmatrix}\mathbf{\Phi}_{\mathrm{S},f}^{j} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix} \quad \in \mathbb{C}^{C(K+1)\times C(K+1)}. \tag{18}$$

The final enhanced signal $\hat{X}_{t,f}^{j}$ is obtained as follows:

$$\hat{X}_{t,f}^{j} = \left(\mathbf{w}_f^{\mathcal{X},j}\right)^{\mathsf{H}}\hat{\mathbf{Y}}_{t,f}^{\mathcal{X},j} \quad \in \mathbb{C}. \tag{19}$$

The above process in the frontend is fully differentiable while only involving elementary operations such as matrix multiplication and inverse. It can be thus easily implemented in various deep learning frameworks as long as the gradient support for these elementary operations is available.

[3]The estimated ATF in this step has the scale ambiguity problem, which is later solved in Step 4) by normalizing w.r.t. the reference channel.

C. Speech recognition backend

Since the frontend output $\{\hat{\mathbf{X}}^j\}_{j=1}^{J}$ is the time-frequency spectrum, we can directly calculate ASR features based on it:

$$\mathbf{O}^j = \text{MVN-LMF}\left(|\hat{\mathbf{X}}^j|\right) \quad \in \mathbb{R}^{T\times D}, \tag{20}$$

where $\text{MVN-LMF}(\cdot)$ denotes calculating the log Mel-filterbank coefficients with mean and variance normalization. D is the feature dimension.

We adopt the joint connectionist temporal classification (CTC) / attention-based encoder-decoder [51] as the ASR backend. As shown in Fig. 1, it consists of three main submodules: encoder, CTC, and decoder. All submodules are shared when processing different separated streams. Firstly, the encoder transforms the speech feature $\mathbf{O}^j = [\mathbf{o}_1^j, \cdots, \mathbf{o}_T^j]^{\mathsf{T}}$ of each speaker j into a high-level representation $\mathbf{H}^j = [\mathbf{h}_1^j, \cdots, \mathbf{h}_{T'}^j]^{\mathsf{T}} \in \mathbb{R}^{T'\times D}$ $(T' \le T)$ with subsampling along the time frame dimension. Then, each representation is fed into the CTC submodule individually to calculate the pair-wise loss between the CTC predictions and labels. Note that when multiple speakers are present $(J \ge 2)$, the well-known label ambiguity problem [52], [53] arises. Here, we adopt the widely used permutation invariant training (PIT) [53], [54] approach as the permutation solver. That is, the final permutation $\hat{\pi}$ of the separated streams is determined by enumerating all possible permutations and finding the optimal permutation that leads to the smallest CTC loss of all speakers:

$$\hat{\pi} = \underset{\pi\in\mathcal{P}_J}{\arg\min}\sum_{j=1}^{J}\mathrm{Loss}_{\mathrm{ctc}}\left(\mathrm{CTC}\left(\mathbf{H}^{\pi(j)}\right), \mathbf{R}^j\right), \tag{21}$$

where $\mathcal{P}_J$ is the set of all possible permutations on $\{1, \cdots, J\}$, and $\pi(j)$ is the permutation for the j-th separated stream. $\mathbf{R}^j = [\mathbf{r}_1^j, \cdots, \mathbf{r}_L^j]^{\mathsf{T}}$ is the reference token sequence for the j-th speaker, and L is the sequence length. The encoder representations are then rearranged into $\hat{\mathbf{H}}^j = \mathbf{H}^{\hat{\pi}(j)}$ based on the best permutation $\hat{\pi}$, and fed into the decoder:

$$\mathbf{c}_n^j = \mathrm{Attention}\left(\mathbf{e}_{n-1}^j, \hat{\mathbf{H}}^j\right), \tag{22}$$

$$\mathbf{e}_n^j = \mathrm{Update}(\mathbf{e}_{n-1}^j, \mathbf{c}_{n-1}^j, \hat{\mathbf{r}}_{n-1}^j), \tag{23}$$

$$\hat{\mathbf{r}}_n^j \sim \mathrm{Decoder}\left(\mathbf{c}_n^j, \hat{\mathbf{r}}_{n-1}^j\right), \tag{24}$$

where $\hat{\mathbf{r}}_n^j$ is the generated output token at the n-th decoding step, with $\hat{\mathbf{R}}^j = [\hat{\mathbf{r}}_1^j, \cdots, \hat{\mathbf{r}}_L^j]^{\mathsf{T}}$ the final decoding output. $\mathbf{c}_n^j$ denotes the context vector obtained from the attention mechanism. $\mathbf{e}_n^j$ is the decoder hidden state. During training, we adopt the teacher-forcing strategy by replacing the history information $\hat{\mathbf{r}}_{n-1}^j$ in Eqs. (23) and (24) with the corresponding reference label $\mathbf{r}_{n-1}^j$.

The final loss function $\mathcal{L}_{\text{e2e}}$ for optimizing the entire system is defined as the combination of two ASR objective functions:

$$\mathcal{L}_{\text{e2e}} = \alpha\mathcal{L}_{\text{ctc}} + (1-\alpha)\mathcal{L}_{\text{att}}\,, \tag{25}$$

$$\mathcal{L}_{\text{ctc}} = \sum_{j=1}^{J}\mathcal{L}_{\text{ctc}}^{j} = \sum_{j=1}^{J}\text{Loss}_{\text{ctc}}\left(\text{CTC}\left(\hat{\mathbf{H}}^{j}\right), \mathbf{R}^{j}\right)\,, \tag{26}$$

$$\mathcal{L}_{\text{att}} = \sum_{j=1}^{J}\mathcal{L}_{\text{att}}^{j} = \sum_{j=1}^{J}\text{Loss}_{\text{att}}\left(\hat{\mathbf{R}}^{j}, \mathbf{R}^{j}\right)\,, \tag{27}$$

where $0 \leq \alpha \leq 1$ is the interpolation factor. While both frontend and ASR backend are optimized solely based on the above ASR loss, it can be shown later in Section V that this fully E2E training scheme can effectively achieve both decent SE performance and strong ASR performance.

D. Attacking the numerical instability issue

In the frontend module, the numerical instability issue [55] has been a well-known problem, which often leads to degraded performance or even failure in speech enhancement. This problem is especially important in the proposed fully E2E approach. Since no explicit signal-level supervision is provided to guide the training of the frontend module, the numerical instability issue would be a major obstacle that prevents the entire system from being well-trained.

The numerical problem in WPE and beamforming generally originates from the complex operations involved in both algorithms, which can be sensitive to the data involved in the operation. For example, the complex matrix inverse is a typical cause of instability, which is prone to large numerical errors when the processed matrix is ill-conditioned or even singular. Such behaviors are particularly undesirable in the joint training with ASR [19], [56], because they can easily cause not-a-number (NaN) gradients in the backward process, which fail to backpropagate correctly and even lead to poor convergence of the entire model [26]. Therefore, in order to improve the numerical stability, we propose to apply the following four complementary approaches to both WPE and beamforming submodules:

(1) Diagonal loading: Diagonal loading [57], [58] is one of the most widely used regularization techniques in conventional beamforming for improving the robustness and numerical stability. It is proven in Chapter 6.6.4 of [39] that diagonally loading the PSD matrix can be regarded as enforcing a quadratic constraint on the MVDR / MPDR beamformer that the norm of the beamformer filter $\left\|\mathbf{w}_f^{\mathcal{X},j}\right\|^2$ is bounded by a constant, which improves the robustness against array perturbations. While there exist various approaches to determining the diagonal loading for specific beamformers [59], [60], we found the following simple strategy is sufficiently working:

$$\mathbf{\Phi}' = \mathbf{\Phi} + \varepsilon\,\text{Trace}(\mathbf{\Phi})\mathbf{I}\,, \tag{28}$$

where $\mathbf{\Phi}$ is any Hermitian matrix to be diagonally loaded. $\mathbf{I}$ is an identity matrix with the same dimension as $\mathbf{\Phi}$. ε is a tiny constant that controls the relative loading level. Note that too large ε will diminish the ability of beamforming to null the weak interference with power less than the loading level. The trace of $\mathbf{\Phi}$ is multiplied in the loading term to make it adaptive to the signal level for better stabilization.

(2) Mask flooring: As described in Section III-B, in our proposed frontend, the matrix inverse operation is applied to a cross-channel PSD matrix such as Eqs. (7)–(8), which is often obtained based on the estimated masks as in Eqs. (9)–(10). Therefore, the masks play an important role in improving the numerical stability in the frontend. In our preliminary experiments, it is noticed that the `MaskNet` can sometimes generate sparse or spiky mask values along certain frequency bins, especially in the early training stage. That means, the `MaskNet` assigns large weights (close to 1) to only a few most relevant time-frequency bins, and tiny weights (close to 0) to the remaining ones. Taking Eq. (10) as an example, the resulting PSD matrix $\mathbf{\Phi}_{\text{N},f}^{j}$, which can be approximated as the weighted sum of d rank-1 matrices with d being a small digit, is likely to be ill-conditioned. Consequently, the WPE / beamforming operation in those frequency bins becomes unstable and may fail frequently. In order to alleviate this problem, we propose to apply a mask flooring operation to the estimated masks from Eq. (4):

$$\hat{\mathbf{M}} = \text{Maximum}(\mathbf{M}, \xi)\,, \tag{29}$$

where $\mathbf{M} \in \{\mathbf{M}_{\text{wpe}}^{j}, \mathbf{M}_{\text{bf-S}}^{j}, \mathbf{M}_{\text{bf-N}}^{j}\}$, and $\hat{\mathbf{M}}$ is the floored mask. ξ is a constant flooring factor. The operator $\text{Maximum}(\cdot,\cdot)$ finds the element-wise maximum between two inputs. As a result, more snapshots of the observation are used (with nonzero weights) in the PSD matrix estimation. The intuition behind this operation is that enough mask values have to be nonzero to overwrite the effect of the constant flooring value. Therefore, `MaskNet` is prevented from learning very sparse or spiky masks, and the stability is potentially improved.

(3) More stable complex matrix operation: There is a rich literature [61]–[63] in approximating the complex matrix inverse using only basic matrix operations. In our initial work [26], we adopted the algorithm in Section 4.3 of [62], which tries to find a factor to construct an invertible real matrix based on the original complex matrix, thus converting the complex matrix inversion into some real matrix operations. However, the success of this algorithm highly depends on the selection of the factor, which can fail in a limited number of trials. Here, a more stable matrix inverse algorithm [61] is implemented, which converts the complex matrix inverse $\mathbf{\Phi}^{-1} \in \mathbb{C}^{m\times m}$ into the inverse of a real matrix $\mathbf{Z} \in \mathbb{R}^{2m\times 2m}$. By replacing each complex matrix with the sum of its real and imaginary parts, the following equation

$$\mathbf{\Phi}\mathbf{\Phi}^{-1} = \mathbf{I} \tag{30}$$

becomes

$$\big(\mathcal{R}\{\mathbf{\Phi}\} + i\mathcal{I}\{\mathbf{\Phi}\}\big)\big(\mathcal{R}\{\mathbf{\Phi}^{-1}\} + i\mathcal{I}\{\mathbf{\Phi}^{-1}\}\big) = \mathbf{I} + i\mathbf{0}\,, \tag{31}$$

where $\mathcal{R}\{\cdot\}$ and $\mathcal{I}\{\cdot\}$ denote the real and imaginary parts of a complex matrix. Thus, the solution to the above equation is equivalent to that of the following system

$$\underbrace{\begin{bmatrix}\mathcal{R}\{\mathbf{\Phi}\} & -\mathcal{I}\{\mathbf{\Phi}\}\\ \mathcal{I}\{\mathbf{\Phi}\} & \mathcal{R}\{\mathbf{\Phi}\}\end{bmatrix}}_{\mathbf{Z}}\begin{bmatrix}\mathcal{R}\{\mathbf{\Phi}^{-1}\}\\ \mathcal{I}\{\mathbf{\Phi}^{-1}\}\end{bmatrix} = \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}\,. \tag{32}$$

Consequently, we can easily derive $\mathbf{\Phi}^{-1}$ as follows:

$$\mathbf{\Phi}^{-1} = \mathbf{Z}^{-1}[0:m, 0:m] + i\,\mathbf{Z}^{-1}[m:2m, 0:m]\,, \tag{33}$$

where $0:m$ and $m:2m$ in each square bracket denote

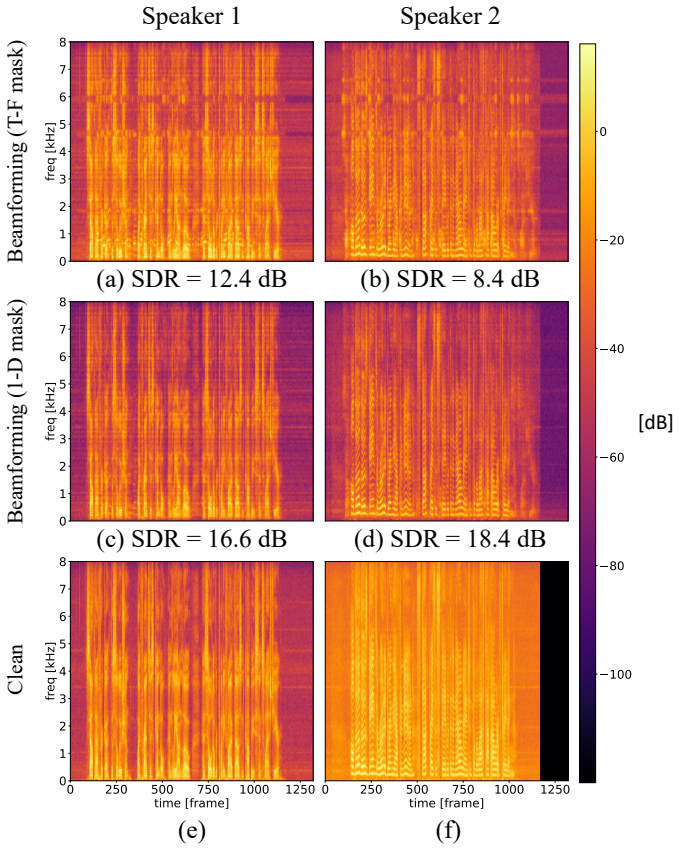


Fig. 2: Frequency permutation problem in the output of the E2E trained frontend. This sample is randomly selected from the evaluation set of spatialized WSJ1-2mix [21], where the energy ratio between speaker 1 and speaker 2 is -4 dB. (a) and (b) correspond to the separated signals of T-F mask-based beamforming. (c) and (d) correspond to VAD-like 1-D mask-based beamforming. (e) and (f) are the clean speech.

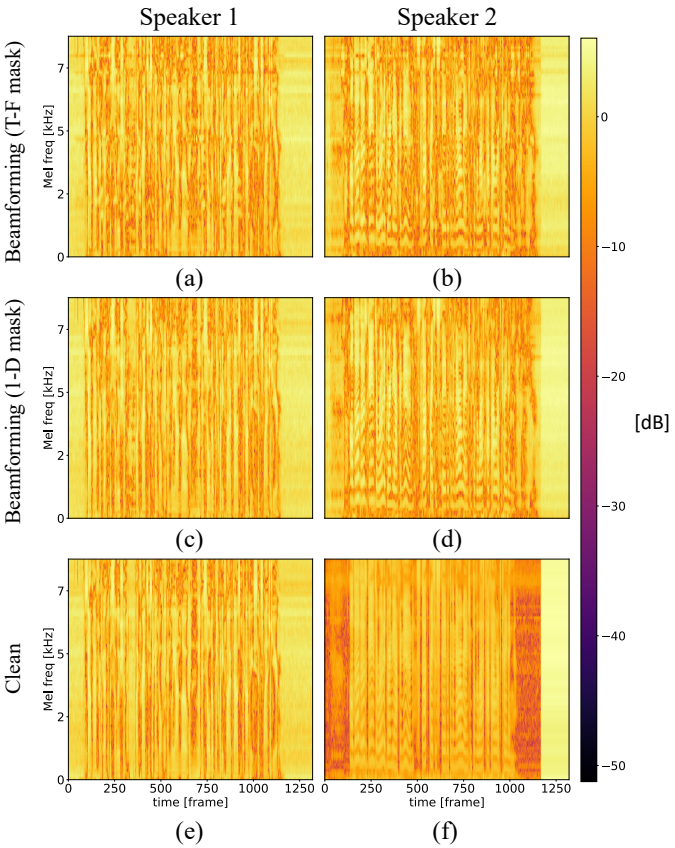


Fig. 3: The log Mel-filterbank features of the separated signals in Fig. 2. (a) and (b) correspond to the separated signals of T-F mask-based beamforming. (c) and (d) correspond to VAD-like 1-D mask-based beamforming. (e) and (f) are the features of the clean speech.

taking the first m rows / columns and the last m rows / columns respectively from the $2m \times 2m$ matrix. i is the imaginary unit. While we can directly calculate each complex matrix inverse term in equations in Section III-B using Eq. (33), it is often unnecessary to perform such explicit computation when it is immediately multiplied with another matrix or vector. That is, when we have the form $\mathbf{\Phi}^{-1}\mathbf{A}$, where $\mathbf{A}$ is either a vector or a matrix, its result $\mathbf{B}$ can be viewed as the solution to the system(s) of linear equations with complex coefficients:

$$\mathbf{\Phi B} = \mathbf{A}\,. \tag{34}$$

It can be similarly converted into a linear equation system with real coefficients:

$$\begin{bmatrix} \mathcal{R}\{\mathbf{\Phi}\} & -\mathcal{I}\{\mathbf{\Phi}\} \\ \mathcal{I}\{\mathbf{\Phi}\} & \mathcal{R}\{\mathbf{\Phi}\} \end{bmatrix} \begin{bmatrix} \mathcal{R}\{\mathbf{B}\} \\ \mathcal{I}\{\mathbf{B}\} \end{bmatrix} = \begin{bmatrix} \mathcal{R}\{\mathbf{A}\} \\ \mathcal{I}\{\mathbf{A}\} \end{bmatrix}. \tag{35}$$

There have been well-established implementations for solving the above real systems of linear equations in the mainstream deep learning toolkits such as `torch.linalg.solve` in PyTorch and `tf.linalg.solve` in TensorFlow. Therefore, we further replace operations of this form in all related equations in Section III-B with the `solve` operation, which can further improve the numerical accuracy and stability.

(4) Double precision: Another major cause of the numerical stability issue is the finite precision of the floating-point numbers used in the frontend processing. Our proposed E2E system by default operates with single-precision data / parameters, which is prone to overflow and underflow in some complex operations. If we increase the precision to double-precision in the frontend processing, the potential numerical error can be reduced in complex operations such as the inverse of a close-to-singular matrix. The overall stability and performance in the frontend processing can be thus improved. Similar effects are also observed in Section 4.4 in [35] when jointly optimizing WPE and acoustic models.

E. Frequency permutation problem

The neural beamformer has shown its capability of separating different speakers when trained with signal-level supervision [64], [65]. However, it is still vulnerable to the well-known frequency permutation problem in the proposed E2E training framework, as shown in Fig. 2. In the separated spectrograms (a) and (b), we can observe several incongruous horizontal patterns such as those around frequencies 4.7 kHz and 6 kHz. The separation results in these frequency bins are apparently misassigned to the wrong speaker according to the clean spectrograms in (e) and (f).

In conventional blind source separation approaches [66], [67], the frequency permutation problem arises due to the separate and independent estimation of the demixing / separation

matrix at each frequency bin. In our case, the beamformer filter for speech separation is also estimated separately for each frequency bin, and the separated spectra are converted into log Mel-filterbank features before the ASR module. Since the frequency information is averaged within each triangle window, which is relatively longer in high frequency bins, the final ASR loss may not well reflect the separation quality of individual frequency bins in the window. To better illustrate this issue, we visualize the extracted log Mel-filterbank features of the separated signals and clean signals in Fig. 3. As we can see, the frequency permutation problem that is clearly shown in Fig. 2 (a) and (b) has been greatly smoothed in the filterbank feature. This largely reduces the impact of permutation errors in local frequency bins, and can be suboptimal for speech enhancement in the frontend when it is the only supervision for training.

In order to alleviate this problem, we propose to replace the time-frequency 2-D masks in `MaskNet` with voice activity detection (VAD) like 1-D masks [56], [68]. That is, only a single mask value is estimated for each time frame, which is then broadcast to all frequency bins in that frame. Since the same mask value is now shared among all frequencies, the frequency permutation problem can be naturally mitigated, as shown in Fig. 2 (c) and (d). Note that the frequency resolution of the estimated masks is sacrificed, thus there is a tradeoff between mitigating the frequency permutation problem and preserving the fine-grained masks for suppressing frequency-specific noise. The overall performance of applying the VAD-like 1-D masks in different scenarios is evaluated in Section V-C.

On the other hand, in some practical scenarios, the microphone array geometry may be known during inference, i.e., the relative positions of all microphones are available. In such cases, it is possible to mitigate the frequency permutation problem in the T-F mask-based beamformer via direction-of-arrival (DOA) estimation. Here, we propose a simple frequency permutation adjustment strategy based on DOA consistency, which does not rely on the ground-truth DOA information of each speaker. First, a magnitude mask $\mathbf{M}^j_{\text{doa}}$ is calculated based on each separated signal $\hat{\mathbf{X}}^j$ after beamforming:

$$\mathbf{M}^j_{\text{doa}} = \frac{|\hat{\mathbf{X}}^j|}{|\mathbf{Y}_b|} \quad \in \mathbb{R}^{T\times F}\,, \tag{36}$$

where $\mathbf{Y}_b \in \mathbb{C}^{T\times F}$ denotes the input signal at the reference channel b. Second, the DOA for each separated signal is estimated, denoted as $\hat{\theta}^j$. Here, we adopt the widely used steered response power phase transform (SRP-PHAT) [69] method for DOA estimation, and enhance it with the estimated time-frequency (T-F) mask $\mathbf{M}^j_{\text{doa}}$ as in [70]:

$$\hat{\theta}^j = \text{SRP-PHAT}(\mathbf{Y}, \mathbf{M}^j_{\text{doa}})\,. \tag{37}$$

Finally, we repeat the DOA estimation process for each single frequency bin f, and obtain the frequency-dependent DOA estimate $\hat{\theta}^j_f$. In each frequency bin f, we calculate the difference between the frequency-dependent DOA estimates for all speakers and the corresponding overall DOA estimates $\hat{\theta}^1, \cdots, \hat{\theta}^J$, which is similar to the PIT process in Eq. (21):

$$\hat{\pi}_{\text{doa},f} = \underset{\pi\in\mathcal{P}_J}{\arg\min} \sum_{j=1}^{J} \left|\text{AngDiff}\left(\hat{\theta}^{\pi(j)}_f, \hat{\theta}^j\right)\right|\,, \tag{38}$$

where $\hat{\pi}_{\text{doa},f}$ is the best permutation of separated signals at the f-th frequency bin that minimizes the overall angular difference. The angular difference function is defined as $\text{AngDiff}(a,b) = (a-b+180^\circ) \mod 360^\circ - 180^\circ$. We can then adjust the frequency permutation according to $\hat{\pi}_{\text{doa},f}$ for each frequency bin. Additionally, we can set a threshold β to restrict the condition for frequency permutation adjustments. That is, the frequency permutation will be adjusted only when

$$\sum_{j=1}^{J} \left|\,\text{AngDiff}(\hat{\theta}^{\hat{\pi}_{\text{doa},f}(j)}_f, \hat{\theta}^j)\right| < \beta\,. \tag{39}$$

This is based on the consideration that a very large angular difference usually indicates poor speech separation performance. In this case, it might be better to stick to the default permutation for the current frequency bin.

F. Strategies for E2E training

In E2E training of the proposed model, there are two main practical issues that need to be resolved. The first is the poor convergence of the model due to lack of intermediate supervision on the frontend module. The second is the massive memory consumption when jointly optimizing both frontend and backend modules. In this section, we propose to solve these problems respectively with the strategies below.

(1) multi-condition training In our prior work [21], [22], it is empirically found difficult to perform straightforward E2E training of frontend and backend modules from scratch. The problem originates from the lack of regularization on individual modules in E2E training. Since the ASR module only observes the separated signals from the frontend, and the frontend is optimized according to the final ASR loss, it is likely for the frontend to learn feature-level enhancement (e.g., Fbank enhancement) instead of signal-level enhancement. Although it is also acceptable if we only care about the ASR performance, such "blind" optimization can easily run into the numerical instability issues introduced in Section III-D and thus leads to poor convergence. Therefore, it is important to regularize the E2E optimization process.

In this paper, we adopt a multi-condition training scheme as in [21], [22] to enforce regularization on the ASR backend. More specifically, in addition to the multi-channel multi-talker training data, we further include auxiliary single-speaker clean data during training. The former is fed into both frontend and backend modules, while the latter is only used to train the ASR backend. As a result, the ASR module learns to obtain good performance on clean data, which in turn rectifies the frontend to generate cleaner signals even without explicit supervision.

(2) memory-efficient training strategies Another challenge in the fully E2E training is the large GPU memory consumption. Speech enhancement models are generally trained with chunked input, which can have much smaller length than the original data to reduce the memory cost. On the other hand, E2E ASR models need to be trained with full-length utterances to match the corresponding transcript. Therefore,

Algorithm 1: Approximated truncated back-propagation through time for E2E training

TBPTT $(\mathbf{Y}, T_{\text{bp}})$
- **inputs:** multi-channel input signal $\mathbf{Y}$; predefined chunk length for truncation T_{bp};
- **output:** separated signals with a partially retained backward graph $\{\hat{\mathbf{X}}^j\}_{j=1}^J$;
- Feed $\mathbf{Y}$ into the frontend w/o backward graph: $\{\hat{\mathbf{X}}^j_{\text{no\_grad}}\}_{j=1}^J \leftarrow \text{StopGradient}\left(\text{Frontend}(\mathbf{Y})\right)$;
- Randomly cut a chunk of length T_{bp} from $\mathbf{Y}$: $\mathbf{Y}_{\text{chunk}} \leftarrow \text{Truncate}(\mathbf{Y}, T_{\text{bp}})$;
- Feed $\mathbf{Y}_{\text{chunk}}$ into the frontend w/ backward graph: $\{\hat{\mathbf{X}}^j_{\text{chunk}}\}_{j=1}^J \leftarrow \text{Frontend}(\mathbf{Y}_{\text{chunk}})$;
- Obtain $\{\hat{\mathbf{X}}^j\}_{j=1}^J$ by overwriting the corresponding part in $\{\hat{\mathbf{X}}^j_{\text{no\_grad}}\}_{j=1}^J$ with $\{\hat{\mathbf{X}}^j_{\text{chunk}}\}_{j=1}^J$;
- **return** $\{\hat{\mathbf{X}}^j\}_{j=1}^J$;

the training data for the proposed E2E model also needs to be full-length utterances, which can cause enormous memory costs either when the input utterance is long or the number of input channels is large. To work around this problem, we propose two alternative training strategies for the E2E model:

a) *Channel sampling*. If there are many input channels ($C > 2$) in the training data, we will randomly select C' channels from the original input to construct a new C'-channel input for training. Here, C' is a relative small number compared to C, so that the memory consumption is largely reduced. This strategy fits well with the `MaskNet` introduced in Section III-B, which estimates masks for each channel independently. Thus, it naturally allows us to use different numbers of input channels during training and evaluation.

b) *Approximated truncated back-propagation through time (TBPTT)*. The approximated truncated back-propagation through time strategy has shown its effectiveness in the joint training of time-domain speech enhancement and ASR models [8], [33]. In this paper, we also verify its efficacy in the context of fully E2E training. The detailed process of the approximated TBPTT strategy is summarized in Algorithm 1. The resulting frontend output only retains the backward graph for a small chunk in the full-length utterance, thus greatly reducing the memory consumption. The full-length output is then fed into the backend module for calculating the ASR loss, which still allows us to jointly optimize both modules effectively. Since our experiments adopt the recurrent neural network (RNN) based `MaskNet` in the frontend module, it can naturally work with this strategy. In addition, as mentioned in [8], the convolutional neural network (CNN) based model is also compatible with this strategy.

In Section V-D, we evaluate both training strategies to illustrate their effectiveness in different conditions.

G. Comparison of training schemes

Although this paper is focusing on the fully E2E training of the proposed model, it can be still trained with various training schemes, as the proposed model is explicitly modularized into frontend and backend modules, as shown in Fig. 1. Therefore, in this section, we would like to discuss the relationship between three training schemes of the proposed model[4], i.e., (1) independent training, (2) fully E2E training, and (3) multi-task learning of different modules.

The training scheme (1) is the most straightforward since it can directly combine pre-trained SE and ASR models without additional efforts. Another advantage of this training scheme is that each module can be designed and updated individually, allowing fast development and reuse of different modules. In many conditions, however, there could be a large mismatch between the resulting frontend and backend modules, since they are often trained on different data. This generally results in severely degraded performance in the testing phase. We will also demonstrate this issue in Section V-E.

The rest training schemes (2) and (3) are very similar, and we can regard (2) as a special case of (3), where only one of the multiple tasks are used. Both training schemes can counteract the mismatch in (1) by jointly optimizing all involved modules during training. The key difference is that (2) only requires labeled data for the final ASR task, while (3) requires supervised data for each involved module. The obvious benefit of (2) is the loosened constraint on data collection, because parallel clean and noisy speech signals are very difficult to collect. On the other hand, it is thus sensitive to the local optima and vulnerable to the frequency permutation problem and numerical instability issues, as no direct constraint is enforced on the frontend module. The training scheme (3) overcomes these problems via the explicit regularization on all involved modules, and generally achieves the best overall performance among the three training schemes. Based on its close connection with (2), we can thus consider multi-task learning as a potential direction to further improve the proposed model when parallel signal-level references are available. Meanwhile, it also provides a reference for the upper bound of fully E2E training in simulation experiments.

IV. Experimental Setup

In Table II, we summarize the four datasets used in our experiments, which can be roughly divided into two different categories: (1) reverberant clean mixtures of $J = 2$ speakers, e.g. spatialized WSJ1-2mix [21] and spatialized WSJ0-2mix [23]; (2) reverberant noisy mixtures of $J = 2$ speakers, e.g. SMS-WSJ [24] and WHAMR! [25].

The multi-condition training strategy in Section III-F is applied when training all E2E models, as it is proven essential for good convergence in the prior work [21]. More specifically, the single-speaker clean speech from the WSJ train_si284 dataset is used for regularizing the ASR backend. In addition, for corpora that contain both anechoic and reverberant versions

[4]One can also perform multi-task learning only on simulated data, while performing the fully E2E training on real data. This can be viewed as the combination of training schemes (2) and (3) in a multi-condition style. For simplicity, however, we only discuss each individual training scheme here.

TABLE II: Detailed information of the corpora used in our experiments. The "max" versions are used for the first four corpora. #Ch denotes the number of channels in the data. T60 denotes the reverberation time. SIR and SNR represent the signal-to-interference ratio and signal-to-noise ratio, respectively.

Dataset	Hours			Sampling Rate (Hz)	#Ch	Condition			
	Train	Dev	Test			T60 (ms)	SIR (dB)	SNR (dB)	Noise
Spatialized WSJ1-2mix [21]	98.5	1.3	0.8	16k	8	[200, 600]	[-10, 10]	-	-
Spatialized WSJ0-2mix [23]	46.9	11.9	7.3	16k	8	[200, 600]	[-10, 10]	-	-
SMS-WSJ [24]	87.4	2.5	3.4	8k	6	[200, 500]	[-5, 5]	[20, 30]	White noise
WHAMR! [25]	58.0	14.7	9.0	16k	2	[100, 1000]	[-10, 10]	[-6, 3]	Real recording

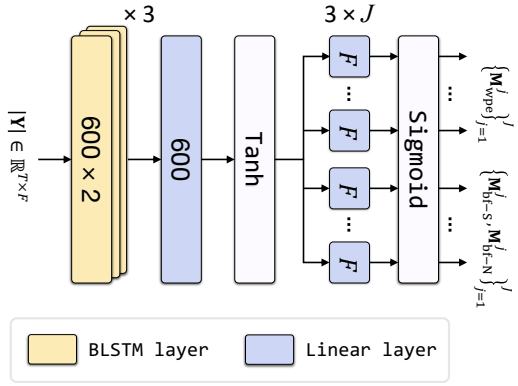


Fig. 4: Architecture of the `MaskNet` in the frontend module.

of data (i.e., spatialized WSJ1-2mix, spatialized WSJ0-2mix, WHAMR!), we further include the anechoic data in the training process to improve the performance. The number of sampled channels in the channel sampling strategy is set to $C' = 2$. The chunk length in the TBPTT training strategy is set to $T_{\text{bp}} = 288$ frames, which is about 3 seconds for 16kHz speech data.

Our experiments were conducted based on the ESPnet toolkit. For all experiments in different corpora, we adopt the following model architectures:

a) Frontend: For 16kHz speech data, the window length and hop length for STFT are 400 and 160, respectively, while the number of discrete Fourier transformer points is 512[5]. For 8kHz speech data, these parameters are respectively 200, 80, and 256. The frequency dimension in the resulting spectrum is thus $F = 257$ and $F = 129$, respectively. The `MaskNet` in Section III is based on the bi-directional long short-term memory (BLSTM) architecture. It consists of 3 BLSTM layers followed by linear projection layers, as illustrated in Fig. 4. The number of hidden units in each layer is also depicted in the figure. It takes as input the magnitude spectrum of the observed signal $|\mathbf{Y}|$, and generates $3 \times J$ estimated masks for WPE and beamforming. As mentioned in Section III-B, each input channel is processed independently, thus yielding $3 \times J$ multi-channel masks. For models based on the VAD-like 1-D masks in Section III-E, the number of hidden units in the final linear projection layers is 1 instead of F, and the generated masks are then repeated F times along the frequency dimension. The number of parameters of the frontend model is 21.99 M. The reference channel b is set to 0 by default for all datasets. Following the setting in [19], the number of filter taps and the delay factor for WPE / WPD are respectively set to $K = 5$ and $\Delta = 3$ during training. The number of iterations is 1 for performing mask-based WPE, and 3 for performing conventional iterative offline WPE (denoted as Nara-WPE [47]). We empirically set the diagonal loading constant ε in Eq. (28) to 10^{-3} and 10^{-8} for WPE and beamforming, respectively. The mask flooring factor ξ in Eq. (29) is empirically set to 10^{-6} and 10^{-2} for WPE and beamforming, respectively. The number of power iterations for RTF estimation in Eqs. (13)–(16) is set to $p = 2$. We found this smaller number of iterations is sufficiently working for the proposed model, which coincides with the observation in [50].

b) Backend: Before the ASR module, we extract 80-dimensional log Mel-filterbank features from each separated stream in the frontend. The ASR module is based on the Transformer architecture, with 12 layers in the encoder and 6 layers in the decoder as in [22]. Each Transformer layer consists of a 4-head self-attention layer with 64 dimensions in each head and a subsequent feedforward layer with 2048 hidden units. Before the Transformer encoder, the log Mel-filterbank features are downsampled by 4 times in the time frame dimension by two convolutional neural network (CNN) layers. Each CNN layer consists of a 3×3 kernel, followed by the `ReLU` activation. The number of feature maps in the first and second CNN layers are 64 and 128, respectively. The number of parameters of the ASR model is 27.14 M.

To compare with the proposed E2E model, we additionally evaluate the performance of a single-channel multi-speaker ASR baseline, denoted by "1-ch 2-spkr ASR". It is also a joint CTC / attention-based encoder-decoder network based on the Transformer architecture [22], [71], whose encoder layers consist of three parts: the mixture encoder, speaker-differentiating (SD) encoder and recognition encoder. We refer the reader to [22] for more detailed introduction of this baseline model. To match the number of parameters in the proposed model, we use 8 SD encoder layers and 14 recognition encoder layers in the baseline model, while other configurations are the same. When training / evaluating the single-channel baseline model, an external Nara-WPE [47] module is used to preprocess the reverberant data. During training, one of the input channels is randomly selected for each sample, so that all channels are likely to be used to train the single-channel baseline after some iterations. In the testing phase, only the reference channel is used for evaluation.

All E2E models are trained using the Adam [72] optimizer

[5]This is a typical window configuration that has been widely used in the ASR task.

TABLE III: 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.

Architecture	WER (%)	PESQ	STOI	SDR (dB)
Original mixture	-	1.20	0.65	-1.45
1-ch 2-spkr ASR	24.86	-	-	-
+ Nara-WPE pre-processing	21.29	-	-	-
Proposed model	16.59	1.30	0.74	2.49
+ (1) Diagonal loading	15.12	**1.32**	**0.75**	**3.25**
+ (2) Mask flooring	16.20	1.30	0.74	2.82
+ (3) Stable complex op.	15.77	1.32	0.75	3.13
+ (4) Double precision	16.43	1.31	0.74	2.87
+ Techs (1)–(4)	**15.01**	1.31	0.74	2.81

with 25000 warmup steps and an initial learning rate of 1.0. We also apply the gradient clipping technique to ensure that the norm of the gradient vector is at most 5. All models are trained for at most 100 epochs, while an early stop will be triggered if no improvement is observed on the development set for 10 epochs. The interpolation factor in the training objective in Eq. (25) is set to $\alpha = 0.2$ during training. The checkpoint at the epoch where the best speech recognition accuracy is obtained on the development set is selected for final evaluation. In the testing phase, the CTC score is also used for joint decoding with a weight of 0.3. Unless otherwise mentioned, an external word-level recurrent neural network language model (RNNLM) [73] trained on the corresponding corpus is used for rescoring.

V. Experimental Results

We used the `pb_bss_eval` package[6] for calculating speech enhancement metrics, including the signal-to-distortion ratio (SDR) [74], short-time objective intelligibility (STOI) [75] and perceptual evaluation of speech quality score (PESQ) [76]. Here, we did not adopt the scale-invariant signal-to-distortion ratio (SI-SDR) [77] as the evaluation metric, because it requires the target signal to be aligned with the input [78], which is hard to achieve in reverberant conditions[7].

A. Evaluation of proposed techniques for improving the numerical stability

We first evaluate the basic performance of the proposed E2E model, where only the essential techniques are applied to ensure a successful training. These techniques include the multi-condition training strategy for good convergence and the channel sampling strategy for fitting in the GPU memory[8] as introduced in Section III-F. The basic performance of the proposed model is shown in the gray row in Table III, where the MVDR beamformer without RTF estimation is used in the frontend. Compared to the single-channel 2-speaker ASR baselines in the second and third row, our proposed model can achieve much better ASR performance, while also being able to provide enhanced speech signals as a byproduct. Next, we evaluate the proposed techniques in Section III-D. The last five rows show the respective performance of applying each individual technique and their combination to the proposed model. It is observed that all techniques lead to better SE performance and thus better ASR performance. This attributes to the improved numerical stability by applying these techniques, as better convergence can be reached under the same model configuration. The combination of all techniques further slightly boosts the ASR performance. Although the final SE performance is slightly degraded compared to that of applying some individual techniques, we would like to emphasize that all four techniques work complementarily to improve the numerical stability, as mentioned in Section III-D. The combination of all techniques makes the E2E training much easier and more stable across various conditions. In addition, it is noticed that the SE performance does not necessarily improve along with the ASR performance. This is also in line with the observation in the previous study on the joint training of SE and ASR models [79]. For the rest experiments below, we apply all four techniques by default to the proposed E2E model.

[6]https://pypi.org/project/pb-bss-eval/0.0.2/

[7]This is even harder when evaluating the E2E trained models, because there is no explicit supervision on the frontend, whose output is thus unlikely to be aligned with a pre-defined target signal.

[8]While either channel sampling or approximated TBPTT can be used here, we opt for the former due to its simplicity. And the performance comparison of these strategies is discussed later.

TABLE IV: Evaluation of different beamformer variants on the spatialized reverberant WSJ1-2mix evaluation set. The best performance is presented for each proposed model by tuning the number of filter taps K and channels C during evaluation.

No.	Model (+ASR)	Formula	WER (%)	PESQ	STOI	SDR (dB)
1	WSJ eval92 [80]	-	4.4	-	-	-
2	MVDR	Eq. (7)	11.66	1.46	0.80	6.48
3	WPE+MVDR		9.50	1.56	0.83	7.73
4	WPE+wMPDR		9.44	1.63	0.82	**8.49**
5	WPD		10.60	1.61	0.82	7.89
6	WPE+MVDR	Eq. (8)	**9.02**	1.50	0.83	6.93
7	WPE+wMPDR		9.23	1.54	0.82	7.12
8	WPD		9.31	1.58	**0.85**	7.91

B. Evaluation of different beamformer variants

Thanks to the flexible design of the frontend module in Section III-B, we can use different numbers of input channels C and filter taps K (in WPE / WPD beamformer) in the evaluation phase, even though the model is trained with a fixed configuration of $C = 2$ and $K = 5$. In this section, we compare the best performance of proposed models based on different beamformer variants, by tuning the configurations of $C \in \{2, 4, 6\}$ and $K \in \{1, 3, 5, 7, 10\}$ for each individual model in the evaluation phase. The best results for each model are tabulated in Table IV. Firstly, the importance of the mask-based WPE component can be shown by comparing rows 2 and 3. Both ASR and SE performances of the E2E model are significantly improved after integrating the WPE module into the beamformer frontend. Secondly, it is shown that the proposed models based on convolutional beamformers (WPE+wMPDR and WPD) achieve better SE performance than those based on the conventional MVDR beamformer, with either Eq. (7) or Eq. (8). This is consistent with the observation in [40], [41], as the convolutional beamformer is designed to be optimal in terms of joint dereverberation and beamforming. Thirdly, comparing rows 3, 4, 5 and rows 6,

TABLE V: Evaluation of different mask types on the spatialized reverberant WSJ1-2mix evaluation set. The best performance is presented for each proposed model by tuning the number of filter taps K and channels C during evaluation. Three beamformer types based on Eq. (7) are evaluated.

No.	Model (+ASR)	Mask	WER (%)	PESQ	STOI	SDR (dB)
1	WPE+MVDR	T-F	9.50	1.56	0.83	7.73
2	WPE+wMPDR	T-F	9.44	1.63	0.82	8.49
3	WPD	T-F	10.60	1.61	0.82	7.89
4	WPE+MVDR	1-D	9.45	1.95	0.86	12.54
5	WPE+wMPDR	1-D	10.26	1.97	0.86	12.20
6	WPD	1-D	10.48	**2.19**	**0.87**	**14.15**

7, 8, the RTF-based beamformer formula, i.e., Eq. (8), tends to yield better ASR results with the proposed E2E model. On the other hand, the SE performance degrades to some extent, which might be caused by the approximation error in the power iteration method in Eqs. (13)–(16). Lastly, since the spatialized WSJ1-2mix evaluation dataset was generated based on the WSJ eval92 subset, the first row in Table IV can be regarded as a topline for our systems. The best WER of the proposed model is only ∼5% worse than this topline, which further illustrates the capability of the proposed model.

C. Evaluation of different mask types

In this section, we evaluate the proposed VAD-like 1-D masks for mitigating the frequency permutation problem. Similar to the last section, in Table V, we compare the best performance of the proposed models based on conventional T-F 2-D masks and VAD-like 1-D masks. As we can see, the proposed models trained with the VAD-like 1-D mask achieve significantly better overall SE performance than the T-F mask based ones on the spatialized WSJ1-2mix data, where no background noise is involved. This result also coincides with the observation in Fig. 2.

On the other hand, however, the ASR performance does not always improve with the proposed VAD-like masks, even though a better SE performance is obtained. This phenomenon has been widely observed in the literature [33], [79], [81], while the cause has not been clearly studied. Here, our conjecture is that the ASR model might learn to ignore some unreliable features from certain frequency bins (e.g., those affected by the frequency permutation problem). This still allows the ASR model to obtain a relatively good performance, while the SE performance is much more sensitive to the frequency permutation problem. To better illustrate this effect, we show the detailed performance comparison of the proposed models with different configurations of C and K in Table VI. It can be observed that the ASR performance with VAD-like 1-D masks is generally worse when the number of input channels C or filter taps K is small. This may originate from the sacrificed frequency resolution due to the VAD-like mask. As mentioned above, the SE and ASR models do not always have consistent improvement, which indicates that the two tasks may favor different optimization directions. For the VAD-like mask based method, since all frequency bins within a time frame share the same mask value, it is harder for `MaskNet` to learn optimal masks for achieving better ASR performance. As C and K increase, the ASR performance with both masks becomes close, because more spatio-temporal information can be exploited to compensate the loss of frequency resolution.

TABLE VI: Performance (Avg. WER [%]) of the proposed models with different mask types and configurations of filter taps K and input channels C on the spatialized reverberant WSJ1-2mix evaluation set. Numbers in brackets are based on VAD-like 1-D masks, while others are based on T-F masks. Three beamformer types (MVDR, wMPDR, and WPD) based on Eq. (7) are evaluated.

	K \ C	2	4	6
WPE+MVDR+ASR	1	16.43 (19.29)	11.03 (12.56)	14.86 (10.94)
	3	15.49 (18.60)	10.57 (11.49)	10.10 (10.13)
	5	15.01 (17.90)	10.29 (11.04)	9.81 (9.87)
	7	14.84 (17.63)	9.75 (10.50)	9.52 (**9.45**)
	10	14.73 (17.39)	9.76 (10.28)	**9.50** (9.51)
	K \ C	2	4	6
WPE+wMPDR+ASR	1	17.51 (20.31)	12.48 (13.02)	11.04 (11.64)
	3	16.38 (19.49)	10.95 (12.08)	9.85 (10.65)
	5	15.35 (18.86)	10.70 (11.76)	9.62 (10.43)
	7	15.48 (18.20)	10.36 (11.72)	**9.44** (**10.26**)
	10	15.11 (18.46)	10.11 (11.42)	10.03 (10.71)
	K \ C	2	4	6
WPD+ASR	1	18.22 (19.11)	11.95 (12.51)	10.86 (10.57)
	3	16.87 (17.72)	11.42 (11.65)	**10.60** (**10.48**)
	5	16.43 (17.12)	11.27 (11.07)	11.06 (10.79)
	7	16.03 (16.98)	11.38 (11.04)	11.05 (11.56)
	10	16.13 (17.00)	11.53 (11.37)	11.67 (13.25)

In order to validate the effectiveness of the proposed method in different scenarios, we further trained and evaluated the proposed models on three additional multi-speaker datasets, i.e., spatialized WSJ0-2mix, SMS-WSJ, and WHAMR!. In these experiments, we only test the MVDR beamformer-based E2E models to simplify the discussion. As shown in Table VII, the proposed models clearly outperform the single-channel two-speaker ASR baselines with Nara-WPE-based preprocessing (2nd row), even when only two input channels are used for evaluation (same as training). Both SE and ASR performances are further improved when all available channels are used for evaluation, which is in line with the observation on the spatialized WSJ1-2mix dataset. Comparing models based on the T-F mask and VAD-like 1-D masks, we can also observe a similar trend that the SE performance improves when applying the VAD-like 1-D masks, except for the WHAMR! corpus. The noise in the WHAMR! corpus consists of real recordings collected in various urban environments (restaurants, cafes, bars, and parks), with a relatively low signal-to-noise ratio (SNR), as shown in Table II. Therefore, the noise energy is usually high and distributed unevenly among different frequency bins, especially for the commonly observed music and babble noise. In such conditions, it can be harmful to use the same mask value for all frequency bins within each frame, as this will inevitably cause noisier estimation of the PSD matrices for certain frequency bins

TABLE VII: Evaluation of the proposed model (WPE+MVDR+ASR) on the evaluation sets of other multi-speaker corpora. The number of filter taps K is set to 5 for all corpora in the evaluation phase.

No.	Model	Mask	Formula	spatialized WSJ0-2mix				SMS-WSJ				WHAMR!			
				WER (%)	PESQ	STOI	SDR (dB)	WER (%)	PESQ	STOI	SDR (dB)	WER (%)	PESQ	STOI	SDR (dB)
1	Original mixture	-	-	-	1.19	0.65	-1.41	-	1.50	0.66	-0.40	-	1.08	0.61	-5.16
2	1-ch 2-spkr ASR + Nara-WPE	-	-	34.73	-	-	-	34.95	-	-	-	(68.82)*	-	-	-
3	Proposed model (2 channels)	T-F	Eq. (7)	11.53	1.30	0.74	3.23	23.26	1.57	0.72	1.68	28.89	1.10	0.66	-2.27
4		1-D	Eq. (7)	14.68	1.36	0.75	4.33	25.72	1.69	0.75	4.93	37.57	**1.11**	**0.68**	**-1.83**
5		T-F	Eq. (8)	11.48	1.29	0.74	2.93	22.06	1.56	0.71	1.64	**27.64**	1.09	0.67	-3.09
6		1-D	Eq. (8)	12.31	1.35	0.76	3.93	25.53	1.68	0.75	4.49	31.45	1.09	0.67	-3.18
7	Proposed model (all channels)	T-F	Eq. (7)	7.56	1.54	0.83	8.55	17.23	1.69	0.78	3.93	same as above			
8		1-D	Eq. (7)	7.80	1.85	0.85	**11.85**	17.50	**2.10**	**0.85**	**11.18**				
9		T-F	Eq. (8)	7.22	1.50	0.83	8.14	16.12	1.68	0.77	3.77				
10		1-D	Eq. (8)	**6.59**	**1.84**	**0.86**	11.83	**17.14**	2.05	0.85	10.36				

* The single-channel ASR baseline suffered from overtraining severely on the WHAMR! corpus. Its speech recognition accuracy converges at ∼91% on the development set, while our proposed models can reach 95%.

TABLE VIII: Evaluation of the proposed DOA-consistency-based frequency permutation adjustment strategy on the SMS-WSJ evaluation set. The T-F mask-based beamforming is used. We set $K = 5$ and $C = 6$ in the evaluation phase.

Threshold β	Formula: Eq. (7)				Formula: Eq. (8)			
	WER (%)	PESQ	STOI	SDR (dB)	WER (%)	PESQ	STOI	SDR (dB)
$< 0°$	**17.23**	1.69	0.78	3.93	**16.12**	1.68	0.77	3.77
$30°$	21.67	1.74	0.78	4.67	21.52	1.70	0.76	3.53
$60°$	25.85	1.74	0.77	4.80	24.86	1.70	0.76	3.63
$90°$	28.79	1.74	0.77	4.86	28.05	1.70	0.76	3.75
$120°$	31.95	1.75	0.77	5.00	30.43	1.71	0.76	3.93
$150°$	34.55	1.75	0.78	5.14	34.00	1.71	0.77	4.10
$180°$	38.04	1.76	0.78	5.31	37.87	1.73	0.77	4.32
$210°$	36.83	1.76	0.78	5.35	35.36	1.73	0.77	4.35
$240°$	34.66	1.76	0.78	5.35	33.03	1.73	0.77	4.36
$270°$	31.91	1.76	0.78	5.35	29.90	1.73	0.77	4.37
$300°$	30.48	1.76	0.78	5.36	28.30	1.73	0.78	4.37
$330°$	29.18	1.76	0.78	5.36	27.08	1.73	0.78	4.38
$360°$	28.67	**1.76**	**0.78**	**5.37**	26.52	**1.73**	**0.78**	**4.38**

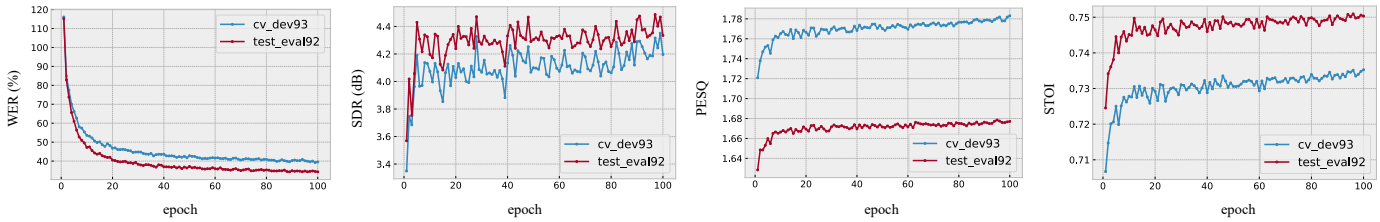
that are dominated by the noise. In addition, while we can observe some relative SDR improvement compared to the original mixture on the WHAMR! corpus, the absolute SE performance is still very poor. Our conjecture is that the small number of channels ($C = 2$) limits the capability of the E2E trained frontend, and the relatively low SNRs also increase the difficulty in such conditions.

Finally, we evaluate the proposed DOA-estimation-based strategy for alleviating the frequency permutation problem with the T-F mask-based beamforming. Note that the strategy is only applied during inference, and the same model as above is used for evaluation. Since this method requires the microphone array geometry to be known in advance, we select the 6-ch test data from the SMS-WSJ corpus for evaluation[9], where all data are simulated based on a 6-mic circular array with radius 10 cm. Table VIII lists the performance when different thresholds β in Eq. (39) are used. Note that when $\beta < 0°$, the results are equal to the previous ones we obtained (Nos. 7 and 9 in Table VII). We can see that the proposed strategy can significantly improve the PESQ and SDR scores in most threshold settings for beamforming with different formulas, especially for the formula defined in Eq. (7). When a larger threshold β is used, the PESQ and SDR scores tend to be further improved, but gradually converge as β approaches $360°$. The STOI score is also slightly improved as β becomes larger. On the other hand, the ASR performance is severely degraded for all settings with $\beta > 0°$. This is likely attributed to the mismatch between training and inference, as the frequency permutations are only adjusted during inference. In addition, the improved SE performance still largely lags behind the VAD-like 1-D mask-based results in Table VII (Nos. 8 and 10). This indicates that the estimated beamforming masks are not optimal even if the frequency permutation is corrected. Therefore, it might be better to directly constrain the estimated masks during training to improve the consistency of frequency permutations, and we would like to investigate it in our future work.

[9]In contrast, other corpora we used above randomly sampled the microphone positions during simulation, thus the array geometry is not fixed.

D. Evaluation of different memory-efficient training strategies

In this section, we validate the effectiveness of the proposed memory-efficient training strategies with MVDR beamformer-based E2E models on the SMS-WSJ corpus. Table IX shows the peak allocated GPU memory when different strategies are used to train the proposed model on the longest samples in SMS-WSJ (around 24s). It can be clearly seen that both proposed training strategies can effectively reduce the GPU memory consumption by about half compared to the plain training with full 6-channel data. Table X further presents the performance of these training strategies, where the first four rows are copied from Table VII, as they used the same configuration. The last four rows show the performance of another proposed strategy in Section III-F, i.e., approximated TBPTT. It can be seen that although only a small portion of



(a) WER (b) SDR (c) PESQ (d) STOI

Fig. 5: Evolutionary performance (WER, SDR, PESQ, and STOI) of the proposed fully E2E trained model (WPE+MVDR+ASR) after each epoch on SMS-WSJ development (`cv_dev93`) and evaluation (`test_eval92`) sets. The model at the 97-th epoch corresponds to the 6-th row of the SMS-WSJ section in Table VII.

TABLE IX: Peak allocated GPU memory when applying different training strategies with batch size 1. The training samples here are all around 24s.

Strategy	Mask	Formula	Mem. (GB)
Plain training (full 6-ch)	T-F	Eq. (7)	4.484
	1-D	Eq. (7)	4.057
	T-F	Eq. (8)	4.889
	1-D	Eq. (8)	4.484
Channel sampling (2-ch)	T-F	Eq. (7)	2.538
	1-D	Eq. (7)	2.383
	T-F	Eq. (8)	2.366
	1-D	Eq. (8)	2.393
Approx. TBPTT	T-F	Eq. (7)	2.019
	1-D	Eq. (7)	2.000
	T-F	Eq. (8)	2.013
	1-D	Eq. (8)	1.996

TABLE X: Evaluation of the proposed memory-efficient training strategies on the SMS-WSJ evaluation set. We set $K = 5$ and $C = 6$ in the evaluation phase.

Strategy	Mask	Formula	WER (%)	PESQ	STOI	SDR (dB)
Channel sampling	T-F	Eq. (7)	17.23	1.69	0.78	3.93
	1-D	Eq. (7)	17.50	**2.10**	**0.85**	**11.18**
	T-F	Eq. (8)	16.12	1.68	0.77	3.77
	1-D	Eq. (8)	17.14	2.05	0.85	10.36
Approx. TBPTT	T-F	Eq. (7)	18.36	1.71	0.76	4.30
	1-D	Eq. (7)	16.32	2.09	0.84	10.61
	T-F	Eq. (8)	**15.94**	1.79	0.78	4.30
	1-D	Eq. (8)	18.58	2.04	0.83	9.15

the input signal is used for backpropagation in the frontend module, the approximated TBPTT strategy can still attain similar ASR and SE performance to the channel sampling strategy. This indicates that we can flexibly select the training strategy accordingly. For example, the approximated TBPTT strategy is favorable when many training samples are very long that cannot fit into the memory even with only 2 channels. Otherwise, channel sampling can be adopted, which is simpler and can be used with any network architectures.

E. Comparison of training schemes of the proposed model

In this section, we compare the aforementioned three training schemes in Section III-G. We follow a similar experimental setup to Section V-D, while the proposed models are trained based on different training schemes. For the independent training and multi-task learning schemes, we additionally include the parallel single-speaker clean speech as the signal-level label for training the frontend module. The convolutive transfer function invariant SDR (CI-SDR) [78] criterion is adopted as

TABLE XI: Comparison of different training schemes on the SMS-WSJ evaluation set. We set $K = 5$ and $C = 6$ in the evaluation phase.

Training scheme	Mask	Formula	WER (%)	PESQ	STOI	SDR (dB)
(1) Independent training	T-F	Eq. (7)	42.30	2.08	0.83	11.88
	1-D	Eq. (7)	37.30	**2.13**	0.85	11.95
	T-F	Eq. (8)	38.20	2.08	0.84	11.38
	1-D	Eq. (8)	40.30	2.04	0.84	10.90
(2) Fully E2E training	T-F	Eq. (7)	17.23	1.69	0.78	3.93
	1-D	Eq. (7)	17.50	2.10	0.85	11.18
	T-F	Eq. (8)	16.12	1.68	0.77	3.77
	1-D	Eq. (8)	17.14	2.05	0.85	10.36
(3) Multi-task learning	T-F	Eq. (7)	15.52	1.98	0.83	10.86
	1-D	Eq. (7)	**15.36**	2.11	**0.85**	**12.14**
	T-F	Eq. (8)	17.15	1.82	0.81	7.81
	1-D	Eq. (8)	15.69	2.07	0.85	11.43

the frontend loss, and the source signal of each speaker is used as the reference. The ASR module in the independent training scheme is trained on the clean WSJ train_si284 dataset. For the multi-task learning scheme, the SE and ASR losses are linearly combined with equal weights to obtain the final objective.

The experimental results are shown in Table XI. (1) We first show the performance of the systems composed of independently trained frontend and backend modules in the first four rows. All systems achieve similarly strong SE performance, which are close to the reported SDR performance (12.9 dB) with oracle ideal binary masks (IBM) in [24]. However, the ASR performance is very poor, with very high WERs on the evaluation set. This attributes to the mismatch between independently trained SE and ASR modules. The ASR module only sees clean speech without reverberation and noise during training, while the beamformer-based SE module inevitably generates imperfect outputs that contain residual noise. Such mismatch thus leads to severe ASR performance degradation. (2) In comparison, our proposed fully E2E training scheme yields reasonably good performance for both speech enhancement and recognition. Since both modules are jointly optimized based on the ASR criterion, the mismatch is largely mitigated. The proposed models with VAD-like 1-D masks can even achieve very similar SE performance to the independently trained frontend module. This further illustrates the effectiveness of the proposed method. (3) In the last four rows, the multi-task learning scheme shows that the performance can be further improved by combining both SE and ASR training criteria. The frontend module is trained to optimize the ASR criterion, while also explicitly guided by the signal-level supervision. It is thus much more stable than the fully E2E training scheme, and achieves much better

SE performance for the models with T-F masks (∼7 dB SDR improvement). Moreover, the ASR performance of all four types of systems is also improved. Overall, we can see that both proposed models trained with (2) and (3) can achieve strong ASR performance, and multi-task learning can further improve the overall performance when parallel signal-level references are available. This observation validates the efficacy of the proposed model and training schemes in the noisy and reverberant condition. In addition, it is also noted that the training scheme (3) is the only one that achieves proportionate performance improvement in both speech separation and ASR. This illustrates the benefit of providing supervision to both frontend and backend modules, and indicates that there is still room for improvement in the proposed fully E2E training scheme.

Finally, in order to better illustrate the training process of the proposed fully E2E model, we manually evaluate the SE and ASR performance of all checkpoints after each epoch. The resultant curve on the SMS-WSJ corpus is shown in Fig. 5. Note that we use the same configuration (e.g., $C = 2$ and $K = 5$) as in training, and unlike in Table VII, no external language model is used when evaluating WERs to better demonstrate the evolution of the acoustic modeling capability. While the ASR performance is initially very poor due to the flat start, it is interesting to see that the SE performance increases very fast and reaches the same level as its final performance after only a few epochs. After the first 8 epochs, the SE performance starts to fluctuate and increases slowly. This phenomenon indicates that the speech enhancement task has a much faster convergence speed than the ASR task, and that the E2E training scheme tends to firstly improve the frontend module so that it can provide relatively stable outputs for the downstream ASR module. Fig. 5 provides an intuitive view of how fully E2E training works in the multi-speaker ASR task, and may inspire further application of this training scheme such as fast adaption of the frontend module in a new domain.

VI. Conclusion

In this work, we present an E2E multi-channel multi-speaker ASR model in noisy and reverberant conditions. The proposed model is composed of a neural beamformer-based frontend and an E2E ASR backend, which is E2E optimized solely based on the final ASR criterion. Several techniques and training strategies are proposed to improve the numerical stability and convergence performance of the E2E model. Extensive experiments on existing multi-channel benchmark datasets have been conducted to validate the efficacy of the proposed method on various conditions. The proposed model is shown to work well with various beamformer types and can achieve competitive performance even in noisy and reverberant conditions, with over 30% relative WER reduction over the single-channel baseline systems. Detailed comparison and performance analyses are also given to better understand the proposed method. Finally, the relationship of the fully E2E training scheme with other existing training schemes is also discussed. In future work, we would like to investigate the E2E training scheme for sparsely overlapped conditions such as conversational speech, which is more realistic in daily communication. More advanced network architectures in the frontend and backend modules will also be explored.

Acknowledgment

This work was supported in part by the National Key Research and Development Program of China (Grant No. 2021ZD0201504), in part by the China NSFC projects (Grant No. 62122050 and No. 62071288), and in part by Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102). Part of this work has been started during the JSALT 2020 workshop at JHU, with support from Microsoft, Amazon and Google. We would like to thank Aswin Shanmugam Subramanian, Reinhold Haeb-Umbach, Marc Delcroix, Keisuke Kinoshita, Tsubasa Ochiai, and Naoyuki Kamo for their insightful discussions in the preliminary experiments. We would also like to thank the anonymous reviewers for their great suggestions that improve the quality of this paper.

References

[1] W. Xiong *et al.*, "The Microsoft 2017 conversational speech recognition system," in *Proc. IEEE ICASSP*, 2018, pp. 5934–5938.
[2] T.-S. Nguyen *et al.*, "Super-human performance in online low-latency recognition of conversational speech," in *Proc. Interspeech*, 2021, pp. 1762–1766.
[3] Y. Qian *et al.*, "Single-channel multi-talker speech recognition with permutation invariant training," *Speech Communication*, vol. 104, pp. 1–11, 2018.
[4] S. Watanabe *et al.*, "CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," in *Proc. 6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, 2020, pp. 1–7.
[5] 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.
[6] D. Yu *et al.*, "Recognizing multi-talker speech with permutation invariant training," in *Proc. Interspeech*, 2017, pp. 2456–2460.
[7] S. Settle *et al.*, "End-to-end multi-speaker speech recognition," in *Proc. IEEE ICASSP*, 2018, pp. 4819–4823.
[8] T. von Neumann *et al.*, "End-to-end training of time domain audio separation and recognition," in *Proc. IEEE ICASSP*, 2020, pp. 7004–7008.
[9] Z. Chen *et al.*, "Multi-channel overlapped speech recognition with location guided speech extraction network," in *Proc. IEEE SLT*, 2018, pp. 558–565.
[10] R. Gu *et al.*, "Enhancing end-to-end multi-channel speech separation via spatial feature learning," in *Proc. IEEE ICASSP*, 2020, pp. 7319–7323.
[11] J. Heymann *et al.*, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. IEEE ICASSP*, 2016, pp. 196–200.
[12] H. Erdogan *et al.*, "Improved MVDR beamforming using single-channel mask prediction networks," in *Proc. Interspeech*, 2016, pp. 1981–1985.
[13] 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.
[14] Z. Zhang *et al.*, "ADL-MVDR: All deep learning MVDR beamformer for target speech separation," in *Proc. IEEE ICASSP*, 2021, pp. 6089–6093.
[15] Y. Xu *et al.*, "Joint training of complex ratio mask based beamformer and acoustic model for noise robust ASR," in *Proc. IEEE ICASSP*, 2019, pp. 6745–6749.
[16] T. Ochiai *et al.*, "Multichannel end-to-end speech recognition," in *Proc. ICML*, 2017, pp. 2632–2641.
[17] C. Liu *et al.*, "A unified network for multi-speaker speech recognition with multi-channel recordings," in *Proc. APSIPA ASC*, 2017, pp. 1304–1307.
[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.*, "Speech enhancement using end-to-end speech recognition objectives," in *Proc. IEEE WASPAA*, 2019, pp. 229–233.

[20] M. H. Soni and A. Panda, "Label driven time-frequency masking for robust continuous speech recognition," in *Proc. Interspeech*, 2019, pp. 426–430.

[21] X. Chang *et al.*, "MIMO-Speech: End-to-end multi-channel multi-speaker speech recognition," in *Proc. IEEE ASRU*, 2019, pp. 237–244.

[22] ——, "End-to-end multi-speaker speech recognition with transformer," in *Proc. IEEE ICASSP*, 2020, pp. 6129–6133.

[23] Z.-Q. Wang *et al.*, "Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation," in *Proc. IEEE ICASSP*, 2018, pp. 1–5.

[24] L. Drude *et al.*, "SMS-WSJ: Database, performance measures, and baseline recipe for multi-channel source separation and recognition," *arXiv preprint arXiv:1910.13934*, 2019.

[25] M. Maciejewski *et al.*, "WHAMR!: Noisy and reverberant single-channel speech separation," in *Proc. IEEE ICASSP*, 2019, pp. 696–700.

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

[27] ——, "End-to-end dereverberation, beamforming, and speech recognition with improved numerical stability and advanced frontend," in *Proc. IEEE ICASSP*, 2021, pp. 6898–6902.

[28] T. Nakatani *et al.*, "Blind speech dereverberation with multi-channel linear prediction based on short time Fourier transform representation," in *Proc. IEEE ICASSP*, 2008, pp. 85–88.

[29] Y. Avargel and I. Cohen, "On multiplicative transfer function approximation in the short-time Fourier transform domain," *IEEE Signal Processing Letters*, vol. 14, no. 5, pp. 337–340, 2007.

[30] T. Nakatani *et al.*, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Trans. ASLP.*, vol. 18, no. 7, pp. 1717–1731, 2010.

[31] Z.-Q. Wang *et al.*, "Leveraging low-distortion target estimates for improved speech enhancement," *arXiv preprint arXiv:2110.00570*, 2021.

[32] K. Qian *et al.*, "Deep learning based speech beamforming," in *Proc. IEEE ICASSP*, 2018, pp. 5389–5393.

[33] W. Zhang *et al.*, "Closing the gap between time-domain multi-channel speech enhancement on real and simulation conditions," in *Proc. IEEE WASPAA*, 2021, pp. 146–150.

[34] X. Xiao *et al.*, "A study of learning based beamforming methods for speech recognition," in *CHiME 2016 workshop*, 2016, pp. 26–31.

[35] 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.

[36] R. Haeb-Umbach *et al.*, "Far-field automatic speech recognition," *Proceedings of the IEEE*, vol. 109, no. 2, pp. 124–148, 2020.

[37] K. Sekiguchi *et al.*, "Direction-aware adaptive online neural speech enhancement with an augmented reality headset in real noisy conversational environments," *arXiv preprint arXiv:2207.07296*, 2022.

[38] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[39] H. L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2004.

[40] 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.

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

[42] Y. Xu *et al.*, "Neural spatio-temporal beamformer for target speech separation," in *Proc. Interspeech*, 2020, pp. 56–60.

[43] Z. Ni *et al.*, "WPD++: An improved neural beamformer for simultaneous speech separation and dereverberation," in *Proc. IEEE SLT*, 2021, pp. 817–824.

[44] Z.-Q. Wang *et al.*, "Sequential multi-frame neural beamforming for speech separation and enhancement," in *Proc. IEEE SLT*, 2021, pp. 905–911.

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

[46] L. Drude *et al.*, "Integrating neural network based beamforming and weighted prediction error dereverberation," in *Proc. Interspeech*, 2018, pp. 3043–3047.

[47] ——, "NARA-WPE: A Python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *13th ITG-Symposium*, 2018, pp. 1–5.

[48] S. Markovich *et al.*, "Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals," *IEEE Trans. ASLP.*, vol. 17, no. 6, pp. 1071–1086, 2009.

[49] 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.

[50] A. Krueger *et al.*, "Speech enhancement with a GSC-like structure employing eigenvector-based transfer function ratios estimation," *IEEE Trans. ASLP.*, vol. 19, no. 1, pp. 206–219, 2010.

[51] S. Kim *et al.*, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. IEEE ICASSP*, 2017, pp. 4835–4839.

[52] J. R. Hershey *et al.*, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE ICASSP*, 2016, pp. 31–35.

[53] 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.

[54] M. Kolbaek *et al.*, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Trans. ASLP.*, vol. 25, no. 10, pp. 1901–1913, 2017.

[55] 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.

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

[57] E. Gilbert and S. Morgan, "Optimum design of directive antenna arrays subject to random variations," *Bell System Technical Journal*, vol. 34, no. 3, pp. 637–663, 1955.

[58] F. Vincent and O. Besson, "Steering vector errors and diagonal loading," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 151, no. 6, pp. 337–343, 2004.

[59] W. Liu and S. Ding, "An efficient method to determine the diagonal loading factor using the constant modulus feature," *IEEE Transactions on Signal Processing*, vol. 56, no. 12, pp. 6102–6106, 2008.

[60] M. Pajovic *et al.*, "Analysis of optimal diagonal loading for MPDR-based spatial power estimators in the snapshot deficient regime," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 2, pp. 451–465, 2018.

[61] 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.

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

[63] F. Soleymani, "A fast convergent iterative solver for approximate inverse of matrices," *Numerical Linear Algebra with Applications*, vol. 21, no. 3, pp. 439–452, 2014.

[64] T. Higuchi *et al.*, "Deep clustering-based beamforming for separation with unknown number of sources," in *Proc. Interspeech*, 2017, pp. 1183–1187.

[65] L. Yin *et al.*, "Multi-talker speech separation based on permutation invariant training and beamforming," in *Proc. Interspeech*, 2018, pp. 851–855.

[66] M. Z. Ikram and D. R. Morgan, "A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation," in *Proc. IEEE ICASSP*, vol. 1, 2002, pp. I–881.

[67] H. Sawada *et al.*, "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 530–538, 2004.

[68] 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.

[69] J. DiBiase *et al.*, "Microphone arrays: Signal processing techniques and applications," in *ch. Robust localization in reverberant rooms*. Springer Verlag, 2001, pp. 157–180.

[70] Z.-Q. Wang *et al.*, "Robust speaker localization guided by deep learning-based time-frequency masking," *IEEE/ACM Trans. ASLP.*, vol. 27, no. 1, pp. 178–188, 2018.

[71] X. Chang *et al.*, "End-to-end monaural multi-speaker ASR system without pretraining," in *Proc. IEEE ICASSP*, 2019, pp. 6256–6260.

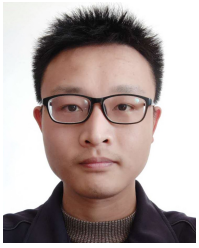
[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[73] T. Hori *et al.*, "End-to-end speech recognition with word-based RNN language models," in *Proc. IEEE SLT*, 2018, pp. 389–396.

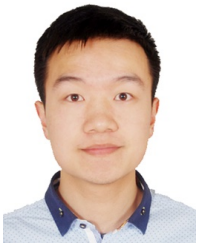
[74] E. Vincent *et al.*, "Performance measurement in blind audio source separation," *IEEE Trans. ASLP.*, vol. 14, no. 4, pp. 1462–1469, 2006.

[75] C. H. Taal *et al.*, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Trans. ASLP.*, vol. 19, no. 7, pp. 2125–2136, 2011.

[76] A. W. Rix *et al.*, "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.
[77] J. Le Roux *et al.*, "SDR—half-baked or well done?" in *Proc. IEEE ICASSP*, 2019, pp. 626–630.
[78] C. Boeddeker *et al.*, "Convolutive transfer function invariant SDR training criteria for multi-channel reverberant speech separation," in *Proc. IEEE ICASSP*, 2021, pp. 8428–8432.
[79] S. E. Eskimez *et al.*, "Human listening and live captioning: Multi-task training for speech enhancement," in *Proc. Interspeech*, 2021, pp. 2686–2690.
[80] S. Karita *et al.*, "A comparative study on transformer vs RNN in speech applications," in *Proc. IEEE ASRU*, 2019, pp. 449–456.
[81] H. Sato *et al.*, "Should we always separate?: Switching between enhanced and observed signals for overlapping speech recognition," in *Proc. Interspeech*, 2021, pp. 1149–1153.



Wangyou Zhang (Student Member, IEEE) received the B.Eng. degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2018. He is currently working toward the Ph.D. degree in Shanghai Jiao Tong University, Shanghai, China, under the supervision of Yanmin Qian. His current research interests include robust speech recognition, speech signal processing and deep learning. In 2021 he pursued a research internship with Microsoft Research Asia, Beijing, China. He was also the recipient of the 2021 MSRA fellowship.



Xuankai Chang (Student Member, IEEE) received the B.Eng. and M.Eng. degrees in 2016 and 2019, respectively, from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. He is currently working toward the Ph.D. degree in the Language Technologies Institute at Carnegie Mellon University, Pittsburgh, PA, USA, advised by Shinji Watanabe. His current research interests include end-to-end speech recognition and the cocktail party problem. He was also the recipient of the Best Paper Award of ASRU in 2019.

Christoph Boeddeker (Student Member, IEEE) received the bachelor's and master's degrees in electrical engineering from Paderborn University, where he is currently working toward the Ph.D. degree, under the supervision of Reinhold Haeb-Umbach. His research interests range from multichannel speech separation, beamforming, and dereverberation to automatic speech recognition on meetings with a focus on combining statistical models and neural networks. In 2017 and 2022 he pursued a research internship with Microsoft Research, Redmond, USA and MERL, Cambridge, USA, respectively.

Tomohiro Nakatani (Fellow, IEEE) received B.E., M.E., and Ph.D. degrees from Kyoto University, Kyoto, Japan, in 1989, 1991, and 2002, respectively. He is a Senior Distinguished Researcher at NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan. Since joining NTT Corporation in 1991, he has been investigating audio signal processing technologies for intelligent human-machine interfaces, including dereverberation, denoising, source separation, and robust ASR. He was a Visiting Scholar at the Georgia Institute of Technology for a year from 2005, and a Visiting Assistant Professor at Nagoya University from 2008 to 2017. He was a member of the IEEE Signal Processing Society (SPS) Audio and Acoustics Technical Committee from 2009 to 2014, and a member of the IEEE SPS Speech and Language Processing Technical Committee from 2016 to 2021.

Shinji Watanabe (Senior Member, IEEE) received the B.S., M.S., and Ph.D. (Dr. Eng.) degrees in 1999, 2001, and 2006, respectively, from Waseda University, Tokyo, Japan. He was a research scientist at NTT Communication Science Laboratories, Kyoto, Japan, from 2001 to 2011, a visiting scholar in Georgia institute of technology, Atlanta, GA in 2009, and a Senior Principal Research Scientist at Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA USA from 2012 to 2017. His research interests include automatic speech recognition, speech enhancement, spoken language understanding, and machine learning for speech and language processing. He has published over 300 papers in peer-reviewed journals and conferences, and received several awards including the best paper award from the IEEE ASRU in 2019. He served as an Associate Editor of the IEEE Transactions on Audio Speech and Language Processing, and was/has been a member of several technical committees including the IEEE Signal Processing Society Speech and Language Technical Committee (SLTC) and Machine Learning for Signal Processing Technical Committee (MLSP).

Yanmin Qian (Senior Member, IEEE) received the B.S. degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2012. Since 2013, he has been with the Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, China, where he is currently a Full Professor. From 2015 to 2016, he also worked as an Associate Research in the Speech Group, Cambridge University Engineering Department, Cambridge, U.K. His current research interests include automatic speech recognition, speaker and language recognition, speech enhancement and separation, key word spotting, and multimedia signal processing. He has published more than 200 papers in peer-reviewed journals and conferences, and received several awards including the best paper award from the IEEE ASRU in 2019. Now he served as a member of IEEE Signal Processing Society Speech and Language Technical Committee (SLTC).