END-TO-END MULTI-SPEAKER ASR WITH INDEPENDENT VECTOR ANALYSIS

Robin Scheibler[1], *Wangyou Zhang*[2], *Xuankai Chang*[3], *Shinji Watanabe*[3], *Yanmin Qian*[2]

[1]LINE Corporation, Tokyo, Japan
[2]X-LANCE Lab, Department of Computer Science and Engineering
MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China
[3]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA
robin.scheibler@linecorp.com

2022 IEEE Spoken Language Technology Workshop (SLT) | 979-8-3503-9690-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/SLT54892.2023.10023037

ABSTRACT

We develop an end-to-end system for multi-channel, multi-speaker automatic speech recognition. We propose a frontend for joint source separation and dereverberation based on the independent vector analysis (IVA) paradigm. It uses the fast and stable iterative source steering algorithm together with a neural source model. Unlike conventional neural beamforming, the number of speakers can be dynamically changed during or after training. The parameters from the ASR module and the neural source model are optimized jointly from the ASR loss itself. We demonstrate competitive performance with previous systems using neural beamforming frontends with only one-ninth of the trainable parameter. First, we explore the trade-offs when using various number of channels for training and testing. Second, we demonstrate that the proposed IVA frontend performs well on noisy data, even when trained on clean mixtures only. Third, we demonstrate recognition of mixtures of three and four speakers with a model trained on mixtures of two only.

Index Terms— end-to-end, multi-speaker, automatic speech recognition, independent vector analysis, multichannel

1. INTRODUCTION

Automatic speech recognition (ASR) technology provides a natural interface for human-to-machine communication [1]. Despite tremendous progress in the last decade, ASR systems are still severely challenged by reverberation, overlapped speech, and noise [2]. Microphone arrays are a powerful tool to fight these degradations. In particular, linear spatial filtering, i.e., beamforming, has been shown to reliably decrease the word error rate (WER) of ASR systems [2]. While optimal beamforming formulations exist [3], e.g. the famous minimum variance distortionless response (MVDR) beamformer, their use has been traditionally limited by the difficulty of estimating the target and noise statistics. These limitations have been recently practically solved by using trained neural networks to estimate these statistics [4]. The resulting neural beamformers (NBF) are highly effective [5]. However, training these networks requires a large amount of parallel speech data, e.g. reverberant mixtures and the isolated anechoic sources they contain. Such data is notoriously difficult to collect. Instead, most works rely on simulation [6, 7]. However, the simulation is often insufficient, and some fully unsupervised approaches have been proposed [8, 9].

The situation for ASR systems is much different since transcripts of actual recordings may be collected by skilled annotators [1]. Indeed, a large amount of annotated speech data has been collected for academic and commercial purposes, e.g., [10, 11, 12]. One can thus bypass the necessity of parallel speech data by cascading enhancement and ASR systems, and training directly from the ASR loss [13, 14]. Building on this approach, an end-to-end (E2E) paradigm for multi-channel, multi-speaker ASR called MIMO-speech [15] has been proposed (see Fig. 1). This approach has demonstrated not only competitive ASR, but also decent separation performance, trained from the ASR loss only. It has been extended to include several advanced joint dereverberation and beamforming methods [16, 17]. Despite all these progresses, the challenge of domain mismatch remains. NBF typically relies on a single input multiple output (SIMO) network to estimate multiple separated source spectrograms, usually two, from a single mixture spectrogram. While NBF can accomodates inputs with different number of channels, the number of output sources is obviously limited to the number of outputs of the SIMO network. In addition, if the test data is sufficiently different from the training data, the SIMO separation network may fail, impeding the beamforming performance.

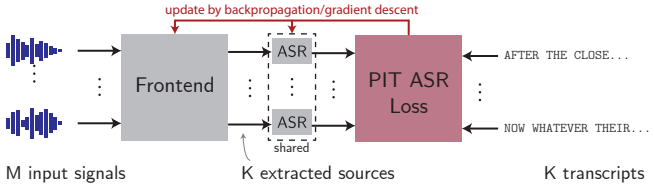

Fig. 1: The MIMO-speech E2E system for K sources and M microphones.

An alternative line of research builds upon independent vector analysis (IVA) [18, 19]. In addition to a statistical model of the sources, their mutual statistical independence is leveraged to help the separation. Vanilla IVA is a blind method, requiring no training data, that can be solved iteratively [20, 21], and rivals sophisticated NBF [22]. Extensions to joint dereverberation and separation have been proposed [23]. In particular, time-decorrelation iterative source steering (T-ISS) [24] is a stable and fast algorithm that avoids matrix inversion. Recently, combining IVA with a neural source model (neural IVA) has attracted attention [25, 26]. One particular approach proposes to train a neural source model end-to-end through T-ISS [21, 27]. Unlike in NBF, the neural source model is a single input single output (SISO) network with parameters shared by all separation outputs. This allows to maintain the high performance of NBF while being agnostic to the number of sources and channels, and robust to a fair amount of data mismatch [27]. The conceptual difference between conventional NBF and neural IVA is illustrated in Fig. 2. On the one hand, T-ISS, as proposed in [24] for joint separation and dereverberation, is limited to the determined case with

978-1-6654-7189-3/22/$31.00 ©2023 IEEE

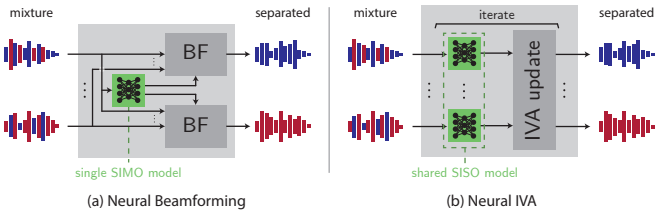

Fig. 2: Illustration of (a) neural beamforming and (b) neural IVA. Gray and green blocks are trainable and non-trainable, respectively.

the same number of sources and microphones. On the other hand, overdetermined ISS [28] is limited to separation only.

In this work, we investigate the use of a T-ISS frontend for MIMO-speech E2E ASR. Our contributions are as follows.

1. We extend T-ISS to the overdetermined case, where more channels than sources are present.
2. We implement the T-ISS frontend in ESPnet [29] and cascade it with an E2E transformer-based ASR backend [30]. The whole system is trained E2E by joint CTC/attention loss [31].
3. In experiments, we explore how the number of channels at training affects test performance (spoiler: more is better). We demonstrate the robustness of T-ISS to mismatch between training and test domains. Finally, we showcase the flexibility of the system by transcribing mixtures of three and four speakers with a system trained on mixtures of two speakers.

2. BACKGROUND

We use the following notation. Bold lower and upper case letters are for vector and matrices, respectively. Furthermore, $\boldsymbol{A}^\top$ and $\boldsymbol{A}^H$ denote the transpose and conjugate transpose, respectively, of matrix $\boldsymbol{A}$. The norm of vector $\boldsymbol{v}$ is $\|\boldsymbol{v}\| = (\boldsymbol{v}^H\boldsymbol{v})^{1/2}$. We index sources, microphones, frequency bands, and time with k, m, f, and n, respectively, running from 1 to K, M, F, and N, respectively. The sets of complex and real positive numbers are $\mathbb{C}$ and $\mathbb{R}_+$, respectively.

2.1. MIMO-Speech and Its Extensions

We are concerned with multi-channel, multi-speaker ASR systems. The MIMO-Speech method [15, 32] proposes a fully end-to-end framework that jointly optimizes the entire system with only the final ASR criterion. The model consists of a beamforming-based frontend for speech separation and an E2E ASR backend. It takes M channels, as inputs, and outputs K text hypotheses, corresponding to K concurrent speakers. First, the frontend extracts K source signals from the input mixture. Second, each extracted signal is processed by the same E2E ASR backend in parallel. This produces K text hypotheses that are evaluated against the K reference transcripts with a utterance-level permutation invariant training (PIT) loss [33]. The process is illustrated in Fig. 1.

2.2. Multichannel Speech Separation and Dereverberation

Physically, the signals from the K sources propagate and reflect on the walls of a room, and mix additively with various amplitudes and time delays at the M microphones. This process can be approximated in the short-time Fourier transform (STFT) domain as follows,

$$\boldsymbol{x}_{fn} = \boldsymbol{A}_f \boldsymbol{s}_{fn} + \boldsymbol{Z}_f \bar{\boldsymbol{x}}_{fn} + \boldsymbol{b}_{fn}, \quad \in \mathbb{C}^M, \tag{1}$$

where $\boldsymbol{s}_{fn} \in \mathbb{C}^K$ is a vector containing the source signals. The matrix $\boldsymbol{A}_f \in \mathbb{C}^{M\times K}$ contains the transfer functions from sources to microphones in its entries. Late reflections from the room are accounted for by $\bar{\boldsymbol{x}}_{fn} = \left[\boldsymbol{x}_{f,n-D}^\top, \ldots, \boldsymbol{x}_{f,n-D-L}^\top\right]^\top$ mixed by the matrix $\boldsymbol{Z}_f \in \mathbb{C}^{M\times ML}$. The reverberation length is given by L, and D is a delay necessary due to the overlap between frames of the STFT. Finally, $\boldsymbol{b}_{fn} \in \mathbb{C}^M$ is a noise term. The role of the frontend is to reduce the second and third terms, and invert $\boldsymbol{A}_f$, if feasible.

2.2.1. Neural Dereverberation and Beamforming

Conventionally, dereverberation and beamforming are done in distinct steps. While several dereverberation methods exist, weighted prediction error (WPE) [34] has been widely adopted for ASR. Ignoring the noise term $\boldsymbol{b}_{fn}$, the dereverberated mixture $\boldsymbol{A}_f\boldsymbol{s}_{fn}$ can be obtained if we know $\boldsymbol{Z}_f$. Provided with a neural network producing a mask r_{fn}^{WPE} hiding the target signal from the spectrogram, the dereverberation filters are given by the minimizer of $\sum_n r_{fn}^{\text{WPE}} \|\boldsymbol{x}_{fn} - \boldsymbol{Z}_f\bar{\boldsymbol{x}}_{fn}\|^2$.

The beamforming filters are computed from the spatial covariance matrices of target speech and noise. Since these are typically not available, a SIMO neural network is trained to estimate masks that extract individual sources from a mixture spectrogram. Let r_{kfn} be the kth source mask. Then, the corresponding spatial covariance matrix is

$$\boldsymbol{\Phi}_{kf} = {}^1\!/_N \sum_n r_{kfn} \breve{\boldsymbol{x}}_{fn} \breve{\boldsymbol{x}}_{fn}^H, \tag{2}$$

where $\breve{\boldsymbol{x}}_{fn}$ is the output of the dereverberation step. Several ways of combining or sharing masks between steps have been proposed and give rise to different NBF such as MVDR, WPD [35], and wMPDR [36]. See [16] for the details. We emphasize here that after the masks are estimated, the beamforming filters are estimated independently. Thus, if the mask estimation fails, it cannot be corrected during the filter computation step.

2.2.2. Independence-based Dereverberation and Separation

This approach to separation builds upon IVA [18, 19]. Unlike, the NBF approach, the foundational hypothesis of IVA is that multiple statistically independent sources are present. The approach has been extended to jointly optimize for dereverberation [23, 24] and include a trainable neural source model [21, 27]. We define the demixing and dereverberation matrices as $\boldsymbol{W}_f \in \mathbb{C}^{M\times M}$ and $\boldsymbol{U}_f \in \mathbb{C}^{M\times ML}$, respectively. For convenience, we concatenate them into a unified dereverberation and separation matrix $\boldsymbol{P}_f = [\boldsymbol{W}_f, \boldsymbol{U}_f] \in \mathbb{C}^{M\times M(L+1)}$. Further let $\tilde{\boldsymbol{x}}_{fn} \in \mathbb{C}^{M(L+1)}$ be the concatenation of $\boldsymbol{x}_{fn}$ and $\bar{\boldsymbol{x}}_{fn}$. The rows of $\boldsymbol{P}_f$ are convolutional beamforming filters but are called *demixing vectors* in the IVA literature. The kth row, i.e. $\boldsymbol{p}_{kf}^H$, extracts the kth target as $y_{kfn} = \boldsymbol{p}_{kf}^H\tilde{\boldsymbol{x}}_{fn}$. Maximum likelihood estimation of $\boldsymbol{P}_f$ can be done by iteratively minimizing

$$\mathcal{L}(\boldsymbol{P}_f) = \sum_{k,f,n} r_{kfn}|\boldsymbol{p}_{kf}^H\tilde{\boldsymbol{x}}_{fn}|^2 - 2\log|\det \boldsymbol{W}_f|. \tag{3}$$

This cost function (3) is derived from the likelihood function of the observed data $\boldsymbol{x}_{fn}$ [23]. The change of variable from $\boldsymbol{x}_{fn}$ to $\boldsymbol{y}_{fn}$ allows to work on the source signals, rather than the mixture, but introduces the log-determinant term. Then, since the sources are independent, the joint probability density function (pdf) is the product of the marginals. Finally, we assume the log-pdf may be majorized by that of the Normal distribution to obtain (3) [20]. The weights

Data: STFT of input channels $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_M$
Result: STFT of separated sources $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_K$
Compute $\mathbf{R}_f, \overline{\boldsymbol{C}}_f$ for all f (see below (8))
for *1 to max. iterations* **do**
  $r_{kfn} \leftarrow u_{fn}(\boldsymbol{Y}_k)$ for all k, f, and n
  for $f \leftarrow 1$ *to* F **do**
    for $\ell \leftarrow 1$ *to* $M(L+1)$ **do**

$$\boldsymbol{g} \leftarrow \begin{cases} \boldsymbol{p}_{\ell f} & \text{if } \ell \leq K \\ \boldsymbol{e}_{\ell} & \text{if } M < \ell \\ [\boldsymbol{J}_f, -\boldsymbol{I}_{M-K}]^H \boldsymbol{e}_{\ell-K+1} & \text{else} \end{cases}$$

      $\boldsymbol{v}_\ell \leftarrow \underset{\boldsymbol{v} \in \mathbb{C}^M}{\arg\min}\, \mathcal{L}(\boldsymbol{P}_f - \boldsymbol{v}\boldsymbol{g}^H)$
      $\boldsymbol{P}_f \leftarrow \boldsymbol{P}_f - \boldsymbol{v}_\ell \boldsymbol{g}^H$
    $\boldsymbol{J}_f \leftarrow$ solution of (8)
    for $n \leftarrow 1$ *to* N **do**
      $(\boldsymbol{Y}_k)_{fn} \leftarrow \boldsymbol{p}_{kf}^H \tilde{\boldsymbol{x}}_{fn}$

Algorithm 1: Pseudocode of Overdetermined T-ISS algorithm.

r_{kfn} are given by a non-linear function $r_{kfn} = u_{fn}(\boldsymbol{Y}_k)$, with $\boldsymbol{Y}_k$ being the $F \times N$ spectrogram the current estimate of source k. In [20], the function $u_{fn} : \mathbb{C}^{F\times N} \to \mathbb{R}_+^{F\times N}$ is derived from the majorization step, and guarantees decrease of the negative log-likelihood. In [21, 27], this exact derivation is abandoned, together with the guarantees, and u_{fn} is defined as a trainable neural network.

While minimization of (3) does not have a closed-form solution, algorithms to efficiently decrease its value exist [23, 24]. T-ISS [24] is particularly suitable for use in E2E training because of low computational cost and lack of matrix inversion. The algorithm proceeds by finding a sequence of $\ell = 1, \ldots, M(L+1)$ optimal rank-1 updates of the form

$$\boldsymbol{P}_f \leftarrow \boldsymbol{P}_f - \boldsymbol{v}_\ell \boldsymbol{p}_{\ell f}^H, \quad \boldsymbol{v}_\ell \leftarrow \underset{\boldsymbol{v} \in \mathbb{C}^M}{\arg\min}\, \mathcal{L}(\boldsymbol{P}_f - \boldsymbol{v}\boldsymbol{p}_{\ell f}^H). \tag{4}$$

For $\ell > M$, we define $\boldsymbol{p}_{\ell f} = \boldsymbol{e}_\ell$, i.e., the vector with all zeros but a one at position ℓ. The closed-form solution for $\boldsymbol{v}_\ell$ in (4) is

$$(\boldsymbol{v}_\ell)_q = \begin{cases} 1 - \left(\sum_n \frac{r_{\ell fn}}{N} |y_{\ell fn}|^2\right)^{-1/2}, & \text{if } q = \ell, \\ \frac{\sum_n r_{qfn} y_{qfn} y^*_{\ell fn}}{\sum_n r_{qfn} |y_{\ell fn}|^2}, & \text{else.} \end{cases} \tag{5}$$

where $y_{qfn} = \boldsymbol{p}_{qf}^H \tilde{\boldsymbol{x}}_{fn}$ for $q \leq M$, and $y_{qfn} = \boldsymbol{e}_\ell^\top \tilde{\boldsymbol{x}}_{fn}$ elsewhere. In contrast to the NBF in [16, 17], spatial cues are taken into account when estimating the source masks. Because the neural network models a single source, the algorithm is easily extended to different numbers of sources. It was also shown to be robust to domain mismatch [27].

The approach shares some ressemblance to NBF. For example, the first term in (3) is a quadratic form involving a weighted spatial covariance matrix similar to (2) with the mask derived by running the source estimate through a neural network. However, whereas NBF uses a single network to estimate all masks, u_{fn} in (3) predicts the remaining noise in a single source estimates. Then, the log-determinant term of (3) pushes the demixing vectors $\boldsymbol{p}_{kf}$ of different sources away from each other.

3. PROPOSED END-TO-END ARCHITECTURE

Our proposed system builds upon the latest methodology of MIMO-Speech [17]. We replace the WPD beamforming frontend by an IVA-based one that performs joint dereverberation and separation. During training, multiple iterations of T-ISS are run to obtain the separation matrix. The same single source mask network is used for all iterations and outputs. The proposed frontend is illustrated in Fig. 3. The system is trained E2E from the ASR loss as illustrated in Fig. 1. Both frontend and ASR backend are fully differentiable, and thus the gradient with respect to the whole system can be computed by backpropagation.

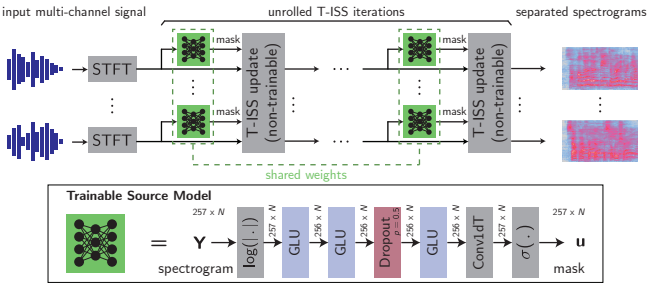

Fig. 3: Block diagram of the proposed frontend and neural source model.

3.1. ASR Transformer Model

We adopt the joint connectionist temporal classification (CTC) and attention-based encoder-decoder [31] as the ASR backend, which consists of four submodules: feature extraction, encoder, CTC, and attention-decoder. The features used are log-Mel filterbank (LMF) features with global mean-variance normalization (MVN). For each separated stream $\boldsymbol{Y}_k$ from the frontend, 80-dimensional log-Mel filterbank features $\mathbf{O}_k$ are firstly extracted via the feature extraction module (MVN-LMF). The extracted feature is then fed into the ASR encoder (Enc) to obtain hidden representations $\mathbf{H}_k$, which are used in both CTC (CTC) and attention-decoder (AttentionDec) submodules for recognition. The ASR procedure is summarized as follows:

$$\begin{aligned} \mathbf{O}_k &= \text{MVN-LMF}(\hat{\mathbf{Y}}_k), & \mathbf{H}_k &= \text{Enc}(\mathbf{O}_k), \\ \hat{\mathbf{R}}_k^{(\text{ctc})} &= \text{CTC}(\mathbf{H}_k), & \hat{\mathbf{R}}_k^{(\text{dec})} &= \text{AttentionDec}(\mathbf{H}_k), \end{aligned}$$

where $\hat{\mathbf{R}}_k^{(\text{ctc})}$ and $\hat{\mathbf{R}}_k^{(\text{dec})}$ are recognition results from CTC and attention-decoder submodules, respectively. We hide the autoregressive nature of the attention-based decoding in the notation. The ASR loss function is constructed based on multi-task learning,

$$\mathcal{L}_{asr} = \alpha \mathcal{L}_{ctc} + (1-\alpha)\mathcal{L}_{dec}, \tag{6}$$

where $0 < \alpha < 1$ is an interpolation factor. The PIT [33] method is applied in the CTC submodule as in [15, 32] to solve the label permutation problem that arises with multiple recognition outputs.

3.2. Overdetermined T-ISS Frontend

Our independence-based frontend is a new extension of the T-ISS [24] algorithm, described in this sub-section, that can be used when more channels are available than there are sources. When there are more channels than sources, i.e., $K < M$, the separation matrix $\boldsymbol{W}_f$ is not square anymore and the algorithm of Section 2.2.2 is not directly applicable. An extension of ISS to the overdetermined case has been proposed, but for separation only [28]. We extend it here to include dereverberation.

First, we write the overdetermined dereverberation and separation operation as a determined system, i.e., square,

$$\begin{bmatrix} \boldsymbol{y}_{fn} \\ \boldsymbol{z}_{fn} \\ \bar{\boldsymbol{x}}_{fn} \end{bmatrix} = \begin{bmatrix} & \boldsymbol{W}_f & & \boldsymbol{U}_f \\ \big[\boldsymbol{J}_f, & -\boldsymbol{I}_{M-N}\big] & & \boldsymbol{0} \\ & \boldsymbol{0} & & \boldsymbol{I}_{ML} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{fn} \\ \bar{\boldsymbol{x}}_{fn} \end{bmatrix}. \tag{7}$$

The separated target sources are $\boldsymbol{y}_{fn} \in \mathbb{C}^K$, and $\boldsymbol{z}_{fn} \in \mathbb{C}^{M-K}$ is a vector of background noise sources to make the system determined. The top part contains $\boldsymbol{P}_f = [\boldsymbol{W}_f\ \boldsymbol{U}_f] \in \mathbb{C}^{K \times M(L+1)}$, of Section 2.2.2, but with K rows, since we only wish to extract so many sources. To complete the separation matrix, we add the strict minimum of parameters $\boldsymbol{J}_f \in \mathbb{C}^{M-K \times K}$. The zeros on the right of the middle block reflect that we do not need to dereverberate the background noise. Note that there is an overload of notation for $\boldsymbol{P}_f$ with respect to Section 2.2.2, but everything coincides when $K = M$, i.e., in the determined case. When $K < M$, the $\log\det$ in (3) contains the top-left $M \times M$ part of the separation matrix in (7). Previous work [37, 38, 39] has shown that a necessary condition for optimality is that the target sources and the noise vector be orthogonal, i.e., $\mathbb{E}\left[\boldsymbol{y}_{fn}\boldsymbol{z}_{fn}^H\right] = \boldsymbol{0}$. From this and (7), we obtain an equation for $\boldsymbol{J}_f$,

$$(\boldsymbol{P}_f \mathbf{R}_f \boldsymbol{E}_1)\boldsymbol{J}_f^H = \boldsymbol{P}_f \mathbf{R}_f \boldsymbol{E}_2, \tag{8}$$

where $\mathbf{R}_f = \mathbb{E}\left[\tilde{\boldsymbol{x}}_{fn}\boldsymbol{x}_{fn}^H\right]$, and $\boldsymbol{E}_1$, $\boldsymbol{E}_2$ are of the appropriate shape, and such that $[\boldsymbol{E}_1\ \boldsymbol{E}_2] = \boldsymbol{I}$. Following the methodology of [28], we update $\boldsymbol{P}_f = [\boldsymbol{W}_f\ \boldsymbol{U}_f]$ and $\boldsymbol{J}_f$ in two steps.

1. We update $\boldsymbol{P}_f = [\boldsymbol{W}_f,\ \boldsymbol{U}_f]$ with (4) and (5), but limit the size of $\boldsymbol{v}_\ell$ to K to match the size of $\boldsymbol{W}_f, \boldsymbol{U}_f$ in (7). We let $\boldsymbol{p}_{\ell f}^H$ be the ℓth row of the matrix in (7) for $\ell > K$.
2. Update $\boldsymbol{J}$ by solving (8).

The resulting algorithm maintains the low-complexity of T-ISS while allowing to use more channels for increased separation power. Pseudo-code is given in Algorithm 1. We note that one matrix inverse is introduced in step 2. However, the size of the matrix to invert is only $K \times K$, with e.g., $K = 2$ for two sources. Despite this small size, we observed some stability issues. The system matrix is not Hermitian symmetric, and its eigenvalues not always positive. Thus, straight diagonal loading, as in [17], does not guarantee stability. Our solution is to replace the $d \times d$ system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ by[1]

$$(\boldsymbol{A}^H \boldsymbol{D}^{-1} \boldsymbol{A} + \epsilon \boldsymbol{I})\boldsymbol{x} = \boldsymbol{A}^H \boldsymbol{D}^{-1} \boldsymbol{b}, \tag{9}$$

where $\boldsymbol{D}$ is a diagonal matrix containing the square norms of the rows of $\boldsymbol{A}$. The system matrix is now guaranteed positive definite. If $\epsilon = 0$, (9) is just the original system multiplied on both sides by $\boldsymbol{A}^H \boldsymbol{D}^{-1}$, and their respective solutions are the same. Normalizing the rows of $\boldsymbol{A}$ with $\boldsymbol{D}$ ensures the sum of the eigenvalues of the system matrix is d. This allows a numerically sensible choice of ϵ.

4. EXPERIMENTS

We conducted several experiments to assess the performance of the proposed method. We investigate the impact of the number of channels, iterations, and the presence of noise.

4.1. Experimental Conditions

We evaluate the proposed method on the WSJ1 corpus [10]. We use two different spatializations of the datasets: $\mathcal{S}_1$ with 8 channels [15], and $\mathcal{S}_2$ with 6 channels [21]. In addition to the clean dataset $\mathcal{C} = \mathcal{S}_1$, we create two noisy datasets. The first ($\mathcal{N}_1$) is obtained by adding noise from CHiME3 [40] to $\mathcal{S}_2$. The second ($\mathcal{N}_2$) is used for mismatched testing and remixes $\mathcal{S}_1$ with simulated diffuse noise [6] created from the TUT environmental sound database [41] with SNR uniformly chosen between $5\,\mathrm{dB}$ to $15\,\mathrm{dB}$. For noisy training, we train on the union of $\mathcal{C}$ and $\mathcal{N}_1$, i.e., $\mathcal{C} \bigcup \mathcal{N}_1$. Unlike previous work [16, 17], we did not do multi-condition training and used only the reverberant mixtures. All the input speech is sampled at $16\,\mathrm{kHz}$. The STFT uses a $25\,\mathrm{ms}$ long Hann window with $10\,\mathrm{ms}$ shifts. The FFT is zero padded to length 512 producing 257-dimensional spectral feature vectors. After the frontend, spectrograms are converted to 80-dimensional log Mel-filterbank features. The training was conducted on an NVidia V100 graphical processing unit (GPU) with $32\,\mathrm{GB}$ RAM.

All the models are implemented in ESPnet [29] using the PyTorch [42] backend. For the baseline, we use the WPD model described in [16]. It uses a bidirectional long-short term memory (BLSTM) network with 600 cells in each direction followed by an output layer producing three masks per target speaker, i.e. 6 in our case. The number of parameters is 23.15 M. WPE is configured with $L = 5$ taps and delay $D = 3$, and runs for a single iteration. The neural source model for T-ISS is the same as in [21]. It has three convolutional layers, with batch-norm, max pooling, and GLU activations. It has a 256 hidden dimension and dropout set to 0.2. Its number of trainable parameters is just 2.57 M. For the models trained on clean data, we preprocessed the input with 5 iterations of AuxIVA-ISS with a non-trainable source model [43]. This is followed by 10 iterations of T-ISS with the neural source model. As it did not seem very effective, we did not apply this preprocessing when training on the noisy dataset. Instead we ran 15 iterations of T-ISS straight. After separation, the scale and phase are aligned to a reference channel by projection back [44]. We used demixing matrix checkpointing [27] for the model to fit on GPU during training.

We used the Adam optimizer with warmup set to 25000 and 50000 steps on the clean and noisy datasets, respectively, and initial learning rate of 1. The WPD baseline was always trained with maximum $M = 8$ channels. We trained multiple T-ISS models on the clean dataset using $M \in \{2, 4, 8\}$ channels. On the noisy dataset, we only trained on $M = 4$ channels due to time constraints. At test time, the number of T-ISS iterations was adjusted to achieve better performance. An external word-level recurrent neural network language model (RNNLM) [45] is applied as shallow fusion in the decoding stage.

Table 1: Performance in terms of WER (%) on the clean test set. Models are trained on clean data with M channels. The four columns on the right are for different number of channels at test time.

Algorithm	M	2ch	4ch	6ch	8ch
Best in [17]†	2ch	15.01	—	9.02	—
WPD	8ch	25.71	12.56	10.00	9.57
T-ISS	2ch	**13.71**	23.40	28.88	31.46
T-ISS	4ch	20.57	10.37	10.86	11.16
T-ISS	8ch	25.71	**9.98**	**9.08**	**9.16**

† Included for reference, training and parameters differ.

4.2. Experimental Results

Effect of Number of Channels Table 1 reports the ASR evaluation results on the clean test set in terms of WER. Each row represents a different trained model. The performance with different numbers of channels at test time is reported in the four right-most columns.

[1] To solve (8), take $\boldsymbol{A} = \boldsymbol{P}_f \mathbf{R}_f \boldsymbol{E}_1$ and $\boldsymbol{b} = \boldsymbol{P}_f \mathbf{R}_f \boldsymbol{E}_2$.

Table 2: Performance on matched and mismatched data. A dagger (†) in the row indicates mismatched conditions between training and testing.

Test set	Algo.	Train set		WER	SIR	SDR	PESQ	STOI
clean ($\mathcal{C}$)	WPD	$\mathcal{C}$		9.57	13.9	**6.9**	**1.88**	**0.855**
	T-ISS	$\mathcal{C}$		**9.16**	**16.8**	3.7	1.78	0.830
noisy1 ($\mathcal{N}_1$)	WPD	$\mathcal{C}$	†	17.12	12.3	8.7	1.70	0.890
	T-ISS	$\mathcal{C}$	†	12.48	**15.6**	6.2	**1.86**	0.913
	WPD	$\mathcal{C} \bigcup \mathcal{N}_1$		**11.40**	14.7	**10.8**	1.79	0.918
	T-ISS	$\mathcal{C} \bigcup \mathcal{N}_1$		11.80	14.4	7.4	1.78	**0.924**
noisy2 ($\mathcal{N}_2$)	WPD	$\mathcal{C}$	†	31.36	6.3	2.7	1.41	0.744
	T-ISS	$\mathcal{C}$	†	**14.55**	**13.7**	2.1	1.45	0.787
	WPD	$\mathcal{C} \bigcup \mathcal{N}_1$	†	15.17	10.0	**5.2**	**1.57**	**0.816**
	T-ISS	$\mathcal{C} \bigcup \mathcal{N}_1$	†	14.75	12.3	2.1	1.43	0.772

We observed that using more channels at training pays off. Models trained this way had lower WERs, even when testing with fewer channels. There is however an exception for T-ISS where the behavior differed if trained with two channels, or more. When trained on two-channel data, the performance was outstanding on two channels test data, even better than the best result from [17], but did poorly with more channels. While not reported due to space constraints, separation metrics increased with the number of channels. This suggests that the ASR backend overfits the artefacts of the separation stage for two channels. Similarly, T-ISS models trained on more channels performed poorly on the two-channel test set. The best performing model was T-ISS trained on 8 channels.

Mismatched Conditions Table 2 reports the ASR performance under different training and test conditions. When trained and tested on clean data, both frontends achieved under $10\,\%$ WER, with T-ISS slightly better at $9.16\,\%$. However, when trained on clean, but tested on noisy data, T-ISS significantly outperformed WPD by $4.6\,\%$. When trained on noisy data, the performance of WPD recovered. T-ISS did about $0.4\,\%$ worse than WPD, but still a little better than in the mismatched condition. Note that in this case, the noise was from the CHiME3 dataset both for training and testing. We thus further tested on the mismatched noisy mixture dataset ($\mathcal{N}_2$). For WPD, the noisy training was effective at improving the robustness, and the WER did not increase as much as before. Again we found T-ISS very robust to mismatch with the lowest WER. Interestingly, its performance was similar regardless of the training data including noise or not. In fact, training on noiseless data had the lowest WER and highest SIR. Table 2 also shows the regular separation metrics SDR, SIR [46], PESQ [47], and STOI [48]. T-ISS had consistently high SIR, but otherwise somewhat lower metrics. The SDR in particular is much lower than that of WPD. This suggests that it achieves good separation, but at the expense of more target degradation.

Separation of 3 and 4 speakers Even though the model was trained on two speaker mixtures, the T-ISS algorithm can be used to separate more, provided that sufficiently many channels are available. We tested this on 3 and 4 speakers mixtures from the noisy dataset using 6 channels. Table 3 shows the results. We note that the problem becomes much harder than in the two speakers case since the per-speaker SNR drops significantly. Still, reasonable ASR performance was maintained in this challenging situation. We compared the performance using clean or noisy training data and found the latter to do better by a few percents.

5. CONCLUSIONS

We have proposed the joint training of a MIMO-speech ASR system with an independent vector analysis frontend using the T-ISS algorithm. T-ISS is an iterative procedure performing joint separation and dereverberation with the help of a neural source model. We demonstrate that E2E training of this system, through the iterations, yields an ASR system robust to data mismatch. The T-ISS frontend trained on clean data only, did best, or at least well enough, on all our test sets. In contrast, the NBF baseline required noisy data in the training set in order to avoid a large performance drop. Furthermore, T-ISS had only one-nineth of the parameters of the baseline.

Table 3: Performance of T-ISS trained with two speakers on mixtures containing $K = 3, 4$ speakers.

K	Train set	WER	SIR	SDR	PESQ	STOI
3	8ch / $\mathcal{C}$	17.80	**10.2**	3.9	**1.52**	0.862
	4ch / $\mathcal{C} \bigcup \mathcal{N}_1$	**16.19**	9.9	**4.8**	1.51	**0.872**
4	8ch / $\mathcal{C}$	33.06	5.8	1.1	**1.34**	0.792
	4ch / $\mathcal{C} \bigcup \mathcal{N}_1$	**30.44**	**6.1**	**2.2**	**1.34**	**0.805**

Another benefit of the T-ISS frontend over conventional NBF is that the number of speakers separated can be dynamically changed without retraining. This is a significant advantage for systems where a variable number of speakers is expected.

Future work should concentrate on the inclusion of a noise model in T-ISS, e.g. [49], to improve the separation metrics. Multi-condition training and curriculum learning are also promising research directions.

6. ACKNOWLEDGMENTS

Yanmin Qian and Wangyou Zhang were supported 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).

References

[1] D. Yu and L. Deng, *Automatic Speech Recognition*, 1st ed., ser. Signals and Communication Technology. London: Springer-Verlag, 2015.

[2] R. Haeb-Umbach *et al.*, "Far-field automatic speech recognition," *Proc. IEEE*, vol. 109, no. 2, pp. 124–148, Feb. 2021.

[3] H. L. Van Trees, *Optimum Waveform Estimation*. John Wiley & Sons, Ltd, 2002, ch. 6, pp. 428–709.

[4] H. Erdogan *et al.*, "Improved MVDR beamforming using single-channel mask prediction networks," in *INTERSPEECH*, Sep. 2016, pp. 1981–1985.

[5] J. Heymann *et al.*, "Neural network based spectral mask estimation for acoustic beamforming," in *ICASSP*, Mar. 2016, pp. 196–200.

[6] E. A. Habets *et al.*, "Generating nonstationary multisensor signals under a spatial coherence constraint," *The Journal of the Acoustical Society of America*, vol. 124, no. 5, pp. 2911–2917, 2008.

[7] R. Scheibler *et al.*, "Pyroomacoustics: A Python package for audio room simulation and array processing algorithms," in *ICASSP*, Apr. 2018, pp. 351–355.

[8] L. Drude *et al.*, "Unsupervised training of a deep clustering model for multichannel blind source separation," in *ICASSP*, May 2019, pp. 695–699.

[9] M. Togami *et al.*, "Unsupervised training for deep speech source separation with Kullback-Leibler divergence based probabilistic loss function," in *ICASSP*, Mar. 2020, pp. 56–60.

[10] Linguistic Data Consortium, and NIST Multimodal Information Group, *CSR-II (WSJ1) Complete LDC94S13A*, Linguistic Data Consortium, Philadelphia, 1994, web Download.

[11] V. Panayotov *et al.*, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, Brisbane, AUS, Apr. 2015, pp. 5206–5210.

[12] D. Galvez *et al.*, "The people's speech: A large-scale diverse english speech recognition dataset for commercial usage," in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmark*, vol. 1, Dec. 2021.

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

[14] T. Ochiai *et al.*, "Unified architecture for multichannel end-to-end speech recognition with neural beamforming," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 8, pp. 1274–1288, 2017.

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

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

[17] W. Zhang et al., "End-to-end dereverberation, beamforming, and speech recognition with improved numerical stability and advanced frontend," in *ICASSP*, Jun. 2021, pp. 6898–6902.

[18] T. Kim *et al.*, "Independent vector analysis: An extension of ICA to multivariate components," in *International conference on independent component analysis and signal separation*. Springer, 2006, pp. 165–172.

[19] A. Hiroe, "Solution of permutation problem in frequency domain ICA, using multivariate probability density functions," in *ASIACRYPT 2016*. Berlin, Heidelberg: Springer, Jan. 2006, vol. 3889, pp. 601–608.

[20] N. Ono, "Stable and fast update rules for independent vector analysis based on auxiliary function technique," in *WASPAA*, Oct. 2011, pp. 189–192.

[21] R. Scheibler and M. Togami, "Surrogate source model learning for determined source separation," in *ICASSP*, Jun. 2021, pp. 176–180.

[22] C. Boeddeker *et al.*, "A comparison and combination of unsupervised blind source separation techniques," in *Speech Communication; 14th ITG Conference*. VDE, 2021, pp. 1–5.

[23] R. Ikeshita *et al.*, "A unifying framework for blind source separation based on a joint diagonalizability constraint," in *EUSIPCO*, Sep. 2019, pp. 1–5.

[24] T. Nakashima *et al.*, "Joint dereverberation and separation with iterative source steering," in *ICASSP*, Jun. 2021, pp. 216–220.

[25] N. Makishima et al., "Independent deeply learned matrix analysis for determined audio source separation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 10, pp. 1601–1615, 2019.

[26] H. Kameoka *et al.*, "Supervised determined source separation with multichannel variational autoencoder," *Neural Computation*, vol. 31, no. 9, pp. 1891–1914, 09 2019.

[27] K. Saijo and R. Scheibler, "Independence-based joint dereverberation and separation with neural source model," in *INTERSPEECH*, Incheon, KR, Sep. 2022, accepted.

[28] Y. Du *et al.*, "Computationally-efficient overdetermined blind source separation based on iterative source steering," *IEEE Signal Process. Lett.*, pp. 1–1, Dec. 2021.

[29] S. Watanabe et al., "ESPnet: End-to-end speech processing toolkit," in *INTERSPEECH*, 2018, pp. 2207–2211.

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

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

[32] X. Chang *et al.*, "End-to-end multi-speaker speech recognition with transformer," in *ICASSP*, 2020, pp. 6129–6133.

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

[34] T. Nakatani *et al.*, "Speech Dereverberation Based on Variance-Normalized Delayed Linear Prediction," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 7, pp. 1717–1731, Sep. 2010.

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

[36] T. Nakatani *et al.*, "Jointly optimal denoising, dereverberation, and source separation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2267–2282, Jul. 2020.

[37] R. Scheibler and N. Ono, "MM algorithms for joint independent subspace analysis with application to blind single and multi-source extraction," *arXiv preprint arXiv:2004.03926*, 2020.

[38] R. Ikeshita *et al.*, "Overdetermined independent vector analysis," in *ICASSP*, 2020, pp. 591–595.

[39] M. Togami and R. Scheibler, "Over-determined speech source separation and dereverberation," in *APSIPA*, Dec. 2020, pp. 705–710.

[40] J. Barker *et al.*, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *ASRU*, Nov. 2015, pp. 504–511.

[41] A. Mesaros *et al.*, "A multi-device dataset for urban acoustic scene classification," in *DCASE*, November 2018, pp. 9–13.

[42] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach *et al.*, Eds., vol. 32. Curran Associates, Inc., 2019.

[43] R. Scheibler and N. Ono, "Fast and stable blind source separation with rank-1 updates," in *ICASSP*, May 2020, pp. 236–240.

[44] N. Murata *et al.*, "An approach to blind source separation based on temporal structure of speech signals," *Neurocomputing*, vol. 41, no. 1-4, pp. 1–24, Oct. 2001.

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

[46] E. Vincent *et al.*, "Performance measurement in blind audio source separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, Jun. 2006.

[47] A. W. Rix *et al.*, "Perceptual evaluation of speech quality (PESQ) — A new method for speech quality assessment of telephone networks and codecs," in *ICASSP*, Salt Lake City, UT, USA, Jan. 2001, pp. 749–752.

[48] A. H. Andersen *et al.*, "A non-intrusive short-time objective intelligibility measure," in *ICASSP*, New Orleans, LA, USA, Mar. 2017, pp. 5085–5089.

[49] Z. Koldovský *et al.*, "Orthogonally-Constrained Extraction of Independent Non-Gaussian Component from Non-Gaussian Background Without ICA," in *Latent Variable Analysis and Signal Separation*. Cham: Springer, 2018, vol. 10891, pp. 161–170.