ESPNET-SE: END-TO-END SPEECH ENHANCEMENT AND SEPARATION TOOLKIT
DESIGNED FOR ASR INTEGRATION

Chenda Li1∗, Jing Shi2,3∗, Wangyou Zhang1∗, Aswin Shanmugam Subramanian3, Xuankai Chang3, Naoyuki Kano, Moto Hira4, Tomoki Hayashi5,6, Christoph Boeddeker7, Zhuo Chen8, Shinji Watanabe8
1Shanghai Jiao Tong University, 2Institute of Automation, Chinese Academy of Sciences, 3Johns Hopkins University, 4Facebook AI, 5Nagoya University, 6Human Dataware Lab. Co., Ltd., 7Paderborn University, 8Microsoft Research

ABSTRACT
We present ESPNet-SE, which is designed for the quick development of speech enhancement and speech separation systems in a single framework, along with the optional downstream speech recognition module. ESPNet-SE is a new project which integrates rich automatic speech recognition related models, resources and systems to support and validate the proposed front-end implementation (i.e. speech enhancement and separation). It is capable of processing both single-channel and multi-channel data, with various functionalities including dereverberation, denoising and source separation. We provide all-in-one recipes including data pre-processing, feature extraction, training and evaluation pipelines for a wide range of benchmark datasets. This paper describes the design of the toolkit, several important functionalities, especially the speech recognition integration, which differentiates ESPNet-SE from other open source toolkits, and experimental results with major benchmark datasets.

Index Terms— Open-source, end-to-end, speech enhancement, source separation, speech recognition

1. INTRODUCTION
As the core parts of the speech processing front-end, speech enhancement and separation (SE) have been studied for decades. Speech enhancement tries to improve the intelligibility and quality of speech contaminated by additive noise and reverberation [1], while speech separation focuses on the speech in multi-speaker conditions, which is also known as the “cocktail party” problem [2]. Speech enhancement and separation are highly desirable for a vast range of applications, such as smart speakers, automatic meeting transcription, automatic captioning for audio/video recordings (e.g., YouTube), multi-party human-machine interaction (e.g., in the world of Internet of things (IoT)), and advanced hearing aids, where noisy speech is commonly encountered [3, 4, 5, 6].

Recently with the great success of deep learning techniques, end-to-end speech enhancement and separation systems (hereinafter collectively referred to as E2E-SE) have grown in popularity and have even started replacing conventional systems described in [1] and [7]. The E2E-SE system, taking the raw waveform of noisy speech as input, outputs one or several speech sources for each target speaker. This end-to-end paradigm can easily integrate signal processing algorithms along with various learnable neural networks, making full use of neural network toolkits. Meanwhile, many downstream speech processing back-end tasks (i.e., automatic speech recognition, keyword spotting and speech translation) can also benefit greatly by cascading after it.

In this paper, we introduce a new E2E-SE toolkit named ESPnet-SE1, which is an extension of the open-source speech processing toolkit ESPnet [8]. ESPnet-SE fully considers the various forms of speech input in the front-end scenes and meanwhile flexibly and organically integrated with the downstream automatic speech recognition (ASR) task, making it a user-friendly toolkit to easily build totally end-to-end robust ASR systems, even without need for clean speech signals. The toolkit provides adaptability to different speech data, including (1) single and multiple speakers, (2) single and multiple channels, (3) anechoic and reverberant conditions. Moreover, thanks to the ripe and efficient ASR modules in ESPnet, rich speech recognition related models, resources and systems can be optionally concatenated after the E2E-SE system, enabling evaluation and joint optimization with ASR, which distinguishes ESPnet-SE from other open source toolkits.

2. RELATED WORK
This section briefly compares ESPnet-SE to other open-source deep learning-based speech enhancement and separation toolkits. We pick up four well-maintained frameworks in this range, including nussl (North-western University Source Separation Library) [9], Onssen (An Open-source Speech Separation and Enhancement Library) [10], Open-Unmix [11] and Asteroid (Audio source separation on Steroids) [12]. Other projects (e.g., padertorch2) may also contain similar features in E2E-SE to some extent, but because they are not systematically designed to address these tasks or still under construction in the paper submission stage, no comparison is made here.

For the first two frameworks nussl and Onssen, they are based on the PyTorch [13] platform and provide several state-of-the-art SE methods, along with the training and evaluation scripts. However, data preparation steps are not provided in the recipes and experiments are not easily configurable from the command line [12]. On the other hand, Open-Unmix and Asteroid provide a whole pipeline from data preparation to evaluation with common line support. Note that the former is aimed for a specific model of Open-Unmix on the music separation task. The same discussion is also applied to a series of isolated speech separation projects, such as kaituoxu/TasNet, kaituoxu/Conv-TasNet, yluo42/TAC and so on. For Asteroid, it is well-designed for the speech separation task, which supports a large range of datasets and architectures, and a set of recipes to reproduce some important papers. However, its overall framework

1The demo page is https://colab.research.google.com/drive/1fjRjCh96SoYLPzPResEFV9Dv4Q2VoIcX1
2https://github.com/fgent/padertorch
The ESPnet-SE toolkit provides two main components: the implementation of different speech enhancement/separation models, and recipes for many common speech enhancement/separation datasets. The ESPnet-SE models are implemented using PyTorch [13] with rich features. The recipes are carefully designed to follow a unified pipeline, with a stage-by-stage processing style. Below we introduce each component in detail.

3. FEATURES OF ESPNET-SE

3.1. Models

Our implemented ESPnet-SE models cover a wide range of scenarios, including (1) time-domain and frequency-domain; (2) single-channel and multi-channel; (3) single-source and multi-source. For time-domain models, we implemented the time-domain audio separation network (TasNet) [21, 22] and the dual-path RNN (DPRNN) [23] based variants. For frequency-domain models, we support both time-frequency (T-F) masking [24] and neural beamformer [25, 26] networks. Among these four models, TasNet and DPRNN operate on the single-channel data, while T-F masking and the neural beamformer can operate on both single-channel and multi-channel data. All models support both speech enhancement for single-source input and speech separation for multi-source input. The input for each model is the raw waveform, and the short-time Fourier transform (STFT) will be used in the frequency-domain models to transform the input to frequency domain. All the processing including STFT is implemented with differentiable PyTorch functions.

T-F masking is a widely used approach to speech enhancement and separation. It estimates a T-F mask for each source in the input, and the mask is used to reconstruct the STFT spectrum of each source. The T-F masking model consists of a recurrent neural network (RNN)\(^3\) followed by several parallel linear layers, with each layer corresponding to one source.

TasNet enhances or separates the audio signal in the time domain. It uses a convolution encoder to encode the audio signal into high-level representations, and then perform enhancement or separation on it. The processed representations are then reconstructed back to audio signals by a deconvolution decoder. Recently, the dual-path RNN (DPRNN) based TasNet [23] achieves state-of-the-art performance on the WSJ0-2mix corpus [27].

The neural beamformer is designed for multi-channel input. It consists of a dereverberation module (optional) and a beamforming module. Each module contains a T-F mask estimation network that

---

\(^3\)We can also easily replace it with other architectures from the ESPnet-ASR module, such as CNN, self-attention and conformer.
is similar to the T-F masking model. The dereverberation module is based on the weighted prediction error (WPE) algorithm [28], and the beamforming module supports various beamforming types, including minimum-variance distortionless response (MVDR) [29], minimum-power distortionless response (MPDR) [30], and weighted power minimization distortionless response (WPD) [31].

3.2. Training for SE and SS
We support multiple loss functions for training the ESPnet-SE models, and they can be divided into three categories: mask approximation loss, signal approximation loss, and metric-based loss.

The mask approximation loss is mainly designed for models with mask estimation components, such as the T-F masking network and the neural beamformer. It measures the error between the estimated mask \( \hat{m} \) and the target mask \( m \). Both mean square error (MSE) and cross-entropy (CE) cost functions can be used. We support various types of target masks, including ideal binary mask (IBM) [32], ideal ratio mask (IRM) [33], ideal amplitude mask (IAM, also known as FFT-mask in [34]) and phase-sensitive mask (PSM) [35].

The signal approximation loss measures the error between the enhanced signal \( \hat{x} \) and the reference signal \( x \). It can be calculated on either the magnitude spectrum or the complex spectrum. A typical cost function for signal approximation is the mean squared error.

The metric-based loss directly measures the quality of the enhanced signal with a specific metric, such as the scale-invariant signal-to-noise ratio (SI-SNR) [21].

Except for the time domain model that does not support the mask approximation loss, all models have implemented different forward functions to support the aforementioned loss functions for training, and it can be easily configured through a configuration file.

For all types of loss functions in speech separation, the permutation invariant training (PIT) [24] is applied to solve the label ambiguity problem. In addition, we also support the unsupervised source separation method from the newly published work [36].

3.3. Evaluation for SE and SS
We evaluate the speech enhancement or separation with various commonly used evaluation metrics, which can be divided into two classes: signal-level metrics and perception-level metrics. For signal-level measurement, the source-to-noise ratio (SNR), source-to-distortion ratio (SDR) [37], source-to-interference ratio (SIR) [37] and source-to-artifact ratio (SAR) [37] are used by default. The scale-invariant signal-to-noise ratio (SI-SNR) [21] is also available. For perception-level measurement, the default metric is the short-time objective intelligibility (STOI) [38] and optionally the perceptual evaluation of speech quality (PESQ) [39] is also available.

Furthermore, the enhanced or separated speech can be evaluated with pretrained single speaker ASR systems, which will be introduced in detail in the following subsection.

3.4. Integration with ASR
Thanks to the unified design, we can easily enable the integration with ASR.

Firstly, the ESPnet-SE models can be directly evaluated with a pretrained ASR model, as illustrated in Fig. 1. The input mixture of \( S \) speakers is first separated by the ESPnet-SE model into \( S \) streams, corresponding to \( S \) different speakers. Then we extract features (e.g. FilterBank) for each stream separately, and feed them into the ASR model to generate the recognized token sequence. Finally, we calculate the word error rate (WER) of the recognition results. When \( S > 1 \), the final WER is determined with the best permutation between recognition results and reference labels, which minimizes the total WER of all speakers.

Furthermore, we can even jointly optimize both ESPnet-SE and the ASR models. The forward process is the same as described above, after replacing WER with the ASR loss. And permutation invariant training is used when \( S > 1 \). Since the forward operations are fully differentiable, the loss can be back-propagated from the ASR model to the ESPnet-SE model, so that the entire system can be jointly optimized.

3.5. Recipes
We provide various all-in-one recipes for several SE benchmark corpora, including WSJ0-2mix [27, 40] and its spatialized version [41], CHiME-4 [19], REVERB [20], DIRHA [42], SMS-WSJ [43], LibriCSS [17], LibriMix [44], WHAM [45] and WHAMR [46]. Note that all these datasets are based on the WSJ [47] or Librispeech [48] corpora, so we can use strong ESPnet-ASR pre-trained models for evaluation and joint optimization. The directory structure of ESPnet-SE recipes is depicted in Fig. 2.

All recipes follow a unified pipeline with a stage-by-stage processing style. The function of each stage in enh.sh is described below:

Stage 1: Corpus-dependent data is prepared in the data directory, following the Kaldi style.

Fig. 1: Block diagram of ESPnet-SE models with ASR integration. The orange part shows the back-propagation process for joint optimization.

Fig. 2: Directory structure of ESPnet-SE.
Table 2: Performance (PESQ / STOI / SDR) on the CHiME-4 1-channel and 6-channel tracks.

<table>
<thead>
<tr>
<th>Model</th>
<th>Track</th>
<th>Dev (Simu)</th>
<th>Test (Simu)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PESQ</td>
<td>STOI</td>
</tr>
<tr>
<td>Noisy Input (CH5)</td>
<td>1-ch</td>
<td>2.17</td>
<td>0.86</td>
</tr>
<tr>
<td>T-F masking</td>
<td></td>
<td>2.55</td>
<td>0.91</td>
</tr>
<tr>
<td>Conv-TasNet</td>
<td></td>
<td>2.46</td>
<td>0.90</td>
</tr>
<tr>
<td>BeamformIt(^7)</td>
<td>6-ch</td>
<td>2.31</td>
<td>0.88</td>
</tr>
<tr>
<td>BLSTM MVDR</td>
<td></td>
<td>2.68</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 3: WER (%) on the CHiME-4 simulation and real data. All results were obtained with the same pretrained ASR model.

<table>
<thead>
<tr>
<th>Track</th>
<th>Model</th>
<th>Dev real</th>
<th>Dev simu</th>
<th>Test real</th>
<th>Test simu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-ch</td>
<td>Baseline [19]</td>
<td>11.6</td>
<td>13.0</td>
<td>23.7</td>
<td>20.8</td>
</tr>
<tr>
<td></td>
<td>ESPnet-ASR + T-F</td>
<td>10.9</td>
<td>12.6</td>
<td>19.5</td>
<td>19.9</td>
</tr>
<tr>
<td></td>
<td>+ Conv-TasNet</td>
<td>15.0</td>
<td>16.4</td>
<td>30.1</td>
<td>26.8</td>
</tr>
<tr>
<td>6-ch</td>
<td>Baseline [19]</td>
<td>5.8</td>
<td>6.8</td>
<td>11.5</td>
<td>10.9</td>
</tr>
<tr>
<td></td>
<td>ESPnet-ASR + T-F</td>
<td>10.9</td>
<td>12.6</td>
<td>19.5</td>
<td>19.9</td>
</tr>
<tr>
<td></td>
<td>+ BeamformIt(^7)</td>
<td>7.3</td>
<td>8.4</td>
<td>13.2</td>
<td>13.9</td>
</tr>
<tr>
<td></td>
<td>+ BLSTM MVDR</td>
<td>5.9</td>
<td>5.3</td>
<td>9.8</td>
<td>8.0</td>
</tr>
</tbody>
</table>

We found this is helpful for training ESPnet-SE models on data without reverberation, as shown in Table 4.

4. EXPERIMENTS EVALUATION

To demonstrate the effectiveness of the proposed ESPnet-SE framework, we conducted several experiments with different datasets from speech enhancement to separation. In this section, we mainly introduce some experimental evaluations on the CHiME-4 speech enhancement challenge, single-channel WSJ0-2mix speech separation task and the spatialized version with multi-channel speech separation. Besides, the speech recognition experiments have also been conducted to show the functionality of our proposed ASR integration.

4.1. Speech enhancement on CHiME-4

We first conduct experiments on CHiME-4 [19], a speech enhancement task where only one speaker exists in each sample. The CHiME-4 dataset contains both simulated data and real recordings, based on the Wall Street Journal (WSJ0) corpus. The sampling rate of CHiME-4 data is 16 kHz. In our experiments, we only use the simulated data for training and evaluation, because there are no clean speech labels for the real recordings. The simulated dataset consists of a 15.1-hour training set, a 2.9-hour development set, and a 2.3-hour test set of 6-channel noisy speech. The 1-channel track of CHiME-4 only selects one of the six channels as input for evaluation, while the 6-channel track uses all channels as input.

We evaluate our implementation of the T-F masking model and TasNet on the 1-channel track and the neural beamformer (denoted as BLSTM MVDR) on the 6-channel track. We also evaluate BeamformIt [49] as a baseline for the 6-channel track. For both tracks, we take the 5-th channel (CH5) clean signal as the reference for computing all metrics, including PESQ, STOI and SDR. The results are shown in Table 2. It shows that our ESPnet-SE models work well on both 1-channel and 6-channel tracks, with significant improvement over the baselines on all metrics.

Furthermore, we also evaluate the speech recognition performance of the ESPnet-SE models with an ASR model, which was pretrained on CHiME-4. The WER results on both simulation and real data are present in Table 3. It can be observed that a simple combination of the neural beamformer and the pretrained ASR model can bring a significant performance improvement, compared to a single ASR model on the 1-channel track. And it also achieves better performance than the baseline model (6-channel track) in [19].

4.2. Speech separation and recognition on WSJ0-2mix

WSJ0-2mix [27] is the current benchmark dataset for the single-channel speech separation task. This dataset consists of a 30-hour training set, a 10-hour validation set, and a 5-hour test set of single-channel two-speaker mixtures without noise and reverberation. In our experiments, we use the WSJ0-2mix data with 8-kHz sampling.

\(^7\)We use the BeamformIt toolkit implemented in https://github.com/xanguera/BeamformIt for evaluation.
Table 4: Performance (PSEQ / STOI / SDRi / WER) on the WSJ0-2mix dataset. WERs are evaluated with the same ASR model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Condition</th>
<th>Dev</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixture</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>uPIT-PSM-relu [6]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>uPIT-PSM-relu + speed perturb</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv-TasNet [22]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv-TasNet + speed perturb</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPRNN [23]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPRNN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Performance (PSEQ / STOI / SDR / WER) on the spatialized WSJ0-2mix dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Condition</th>
<th>Dev</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Mixture</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-ch Deep clustering [41]</td>
<td>anechoic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRM / IBM [41]</td>
<td></td>
<td></td>
<td>12.9</td>
</tr>
<tr>
<td>2-ch BLSTM MVDR</td>
<td>3.48 0.97 20.6</td>
<td>3.43 0.97 20.5</td>
<td>34.4</td>
</tr>
<tr>
<td>Original Mixture</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-ch Oracle MCWF [41]</td>
<td>reverberant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRM / IBM [41]</td>
<td></td>
<td></td>
<td>10.9</td>
</tr>
<tr>
<td>8-ch BLSTM MVDR</td>
<td>2.45 0.81 10.19</td>
<td>2.40 0.82 10.05</td>
<td>70.3</td>
</tr>
<tr>
<td>+ Nara-WPE [50]</td>
<td>2.67 0.83 11.55</td>
<td>2.67 0.84 11.94</td>
<td>56.7</td>
</tr>
<tr>
<td>+ DNN-WPE [10]</td>
<td>2.17 0.75 4.90</td>
<td>2.18 0.78 5.33</td>
<td>-</td>
</tr>
</tbody>
</table>

rate for training and evaluation. With this benchmark, plenty of speech separation methods have been proposed in recent years, including deep clustering [27], PIT [24], TasNet [21, 22], and DPRNN [23].

Among these methods, we compare our implementation of three representative methods in the time-frequency domain (uPIT [6]) and time domain (Conv-TasNet [22] and DPRNN) to those from prior studies. The results are shown in Table 4. As we can see, our implementations achieve promising results on the WSJ0-2mix, with a performance comparable to originally reported.

Interestingly, we also find that speed perturbation [51], a broadly used method for data augmentation in ASR, is also helpful in training the ESPnet-SE model. By applying speed perturbation with speed factors of 0.9, 1.0 and 1.1, we can observe an absolute SDR improvement of 1.2 dB on the TasNet model and 0.4 dB on the T-F masking model.

In addition, we also evaluate these well-trained speech separation models with an ESPnet-ASR model pretrained on WSJ. The WER of the pretrained ASR model on WSJ Dev93 and Eval92 are 9.9% and 7.0% respectively. By simply combining the ESPnet-SE model and ASR model, we obtain a surprisingly good performance on the WSJ0-2mix dataset. As shown in the last column in Table 4, the combination of Conv-TasNet and ASR models achieves 16.7% WER on the evaluation set, and the DPRNN achieves better performance with 16.1% WER.

4.3. Speech separation on spatialized WSJ0-2mix

The spatialized WSJ0-2mix [41] is a multi-channel speech separation dataset, extended from the original WSJ0-2mix dataset. A room impulse response (RIR) generator\(^8\) based on the image method [52] is used to spatialize the WSJ0-2mix dataset. It contains an anechoic and a reverberant version, both of which consists of 8-channel two-speaker mixtures. In our experiments, we use the spatialized WSJ0-2mix data with 8-kHz sampling rate for training and evaluation.

We evaluate our implementation of the neural beamformer (denoted as BLSTM MVDR) on both anechoic and reverberant conditions. For evaluation on both anechoic and reverberant data, we use the anechoic signal of each speaker at the first channel as the reference, and the results are shown in Table 5. We can observe that in the anechoic condition, our implemented neural beamformer outperforms the prior work [41], with more than 7dB SDR improvement. In the reverberant condition\(^9\), our model still achieves a good performance on a par with the prior work [41]. And with an additional WPE module for dereverberation, the performance can be further improved.

Finally, we also evaluate the performance with ASR integration. The same ASR model trained on WSJ is used for ASR evaluation. The results are present in the last column in Table 5.

4.4. Joint training of SE and ASR

As illustrated in Fig. 1, our entire system could be optimized with the final ASR loss, while the SE loss could also be an additive option to train the same architecture. Here, we showed two example experiments on the WSJ0-2mix dataset trained with both SE (speech separation here) and ASR loss. The results are shown in Table 6. The

---

\(^8\)https://github.com/ehabets/RIR-Generator

\(^9\)Note that the results in the reverberant condition is not fully comparable, because a different reference signal is used in [41].

\(^10\)The DNN-WPE module is still under development.
model we use is the same as each module of T-F masking or Conv-TasNet separation network and ASR model used in section 4.2, while it is directly trained with WSJ0-2mix rather than WSJ from scratch.

From the results we could observe that, the joint optimization works well for the downstream application. With the training only on the WSJ0-2mix, comparable or better WER has been obtained compared with the strong single-speaker ASR trained on the whole WSJ dataset (see in Table 4). However, the separation performance gets worse than the isolated separation model. This tendency is also observed from some other works [14, 18].

In addition, we are working on the end-to-end joint training with ASR loss only. Previous studies [15, 53, 54] have shown promising results in this direction, and all of them got the implementation with ESPnet. Some ongoing work by us is to reproduce these works. It is optimistic to make it work since the ESPnet-SE shares the similar design ideas and the bases with ESPNet.

### 4.5. Visualization of enhanced spectrogram

In this section, we visualize the spectrograms of both input and enhanced signals to demonstrate the power of our ESPnet-SE model, which can be easily generated within the ESPnet-SE recipe. Here we adopt the well-trained DPRNN model for separation. We randomly selected two same-gender speech mixtures from the WSJ0-2mix test set. The spectrograms of the mixed signals, the corresponding clean source signals, and the estimated source signals are shown in Fig. 3. As we can see, our implemented DPRNN model can generate very good estimates of the source signals from the mixture.

In this paper, we present an end-to-end speech enhancement and separation toolkit, named ESPnet-SE, which can be tightly integrated with ASR. The toolkit is an extension of the open-source toolkit ESPnet, and supports various state-of-the-art speech enhancement and separation models with rich features. It also provides plenty of all-in-one recipes with a unified stage-by-stage processing pipeline, including data preparation, feature extraction, model training and evaluation. A wide range of benchmark datasets are covered in the current toolkit. A core feature that differentiates this toolkit from others is the capability of ASR integration. With the unified design in ESPnet-SE, we can easily integrate speech enhancement with ASR, such as model evaluation with the ASR criterion and joint optimization with ASR.

In the future, we will support more speech enhancement and separation models, such as MIMO-Speech [15], which is tightly integrated with ASR. We will also support target-speaker speech extraction models like speaker beam [55] in the future. In addition, we will work on a more flexible design to enable the integration of external E2E-SE models with ESPnet-ASR.

### 5. CONCLUSIONS

A part of this work was studied during JSALT 2020 at JHU, with support from Microsoft, Amazon, and Google. Chenda Li and Wangyou Zhang are also supported by the China NSFC project (No. 62071288 and No.U1736202).

### 6. ACKNOWLEDGEMENT

A part of this work was studied during JSALT 2020 at JHU, with support from Microsoft, Amazon, and Google. Chenda Li and Wangyou Zhang are also supported by the China NSFC project (No. 62071288 and No.U1736202)

### 7. REFERENCES


[3] Harald Gustafsson, Sven E Nordholm, and Ingvar Claesson, “Spectral subtraction using reduced delay convolution and...


[35] Hakan Erdogan, John R Hershey, Shinji Watanabe, and...


