SINGLE CHANNEL VOICE SEPARATION FOR UNKNOWN NUMBER OF SPEAKERS UNDER REVERBERANT AND NOISY SETTINGS

Shlomo E. Chazan[1]*, *Lior Wolf*[1,2], *Eliya Nachmani*[1,2], *Yossi Adi*[1]

[1]Facebook AI Research, [2]Tel Aviv University

ABSTRACT

We present a unified network for voice separation of an unknown number of speakers. The proposed approach is composed of several separation heads optimized together with a speaker classification branch. The separation is carried out in the time domain, together with parameter sharing between all separation heads. The classification branch estimates the number of speakers while each head is specialized in separating a different number of speakers. We evaluate the proposed model under both clean and noisy reverberant settings. Results suggest that the proposed approach is superior to the baseline model by a significant margin. Additionally, we present a new noisy and reverberant dataset of up to five different speakers speaking simultaneously.

Index Terms— source separation, speech processing, speaker classification

1. INTRODUCTION

In real-world acoustic environments, a speech signal is frequently corrupted by a noisy environment, room conditions, multi-talker setup, etc. The ability to separate a single voice from multiple conversations is crucial for any speech processing system designed to perform under such conditions. Over the years, many attempts have been made to tackle this separation problem considering single microphone [1, 2], multiple microphones [3, 4], supervised and unsupervised learning [5, 6].

In this work, we focus on fully supervised voice separation using a single microphone, which has seen a great leap in performance following the recent success of deep learning models considering both frequency domain [1, 2, 7, 8, 9, 10], and time-domain [11, 12, 13, 14, 15, 16] modeling.

Despite its success, most prior work assumes the number of speakers in the mixture to be known a-priori. Recently, several studies proposed various methods to tackle this problem. The authors of [17, 18, 19] suggest to separate *one speaker at a time* using a recursive solution. This requires C sequential forward passes to separate C sources and it is not clear when to stop the separating process. The authors of [20] proposed a similar *one speaker at a time* solution however they were mainly interested in automatic speech recognition as the final downstream task. Another line of prior work, optimize the network to output the maximum number of speakers regardless of the actual number of speakers present in the input mixture. At test time, the number of speakers is determined by detecting the number of silent channels [7, 21]. Although this method is shown to perform well, it was evaluated only under an anechoic setup while considering up to three speakers.

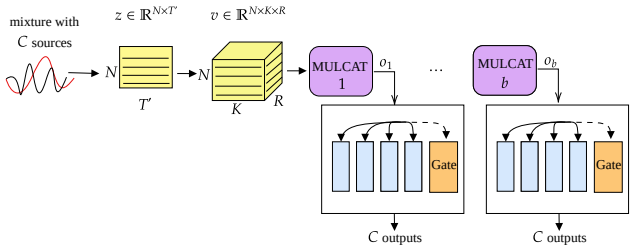

Fig. 1: The architecture of the proposed network. The feature extraction constructed with 1D convolutions and chunking. Then b units are applied using the same separation heads to produce output after each block.

The most relevant prior work to ours is [22]. In this study, the authors suggested training several models, each for separating a different number of speakers. A model selection heuristic is applied on top of the obtained models predictions to detect non-activated channels (noise / silence). Despite its success, it has two main drawbacks. First, several different models were trained separately, hence at test time the input mix is propagating throughout each separately. This makes inference costly in terms of memory and computing power. Additionally, training each model separately does not benefit from shared representations, e.g., the representation learned while separating two speakers can be beneficial for separating four speakers. Second, under the unknown number of speakers setting only anechoic setup was considered. While [22] reported results on WHAMR! dataset [23], which contains noisy reverberant mixtures, this dataset consists of mixtures of two sources only.

In this study, we propose a unified approach to separate up to five different speakers simultaneously speaking using several separation heads together with shared representations. To better handle varying number of speakers in the mixture, we jointly optimize separation and classification of the number

Samples: `https://shlomke.github.io/Samples/ICASSP_2021`. *Work done while Shlomo was an Intern at Facebook AI Research

of speakers in the mixture. Our model is working directly on the raw waveform and was evaluated under both anechoic and noisy reverberant environments. The proposed model obtains superior performance over the baseline methods under both clean and noisy reverberant settings, especially when considering the number of speakers in the mixture to be unknown. We additionally release the scripts used to generate the proposed noisy reverberant datasets.

2. PROBLEM SETTING

2.1. Anechoic room

Consider a single microphone, recording a mixture of C different sources $\boldsymbol{s}^j \in \mathbb{R}^T$, where $j \in [1, \dots, C]$ in an anechoic enclosure where the source length, T can vary. The mixed signal is therefore: $\boldsymbol{x} = \sum_{j=1}^{C} \alpha^j \cdot \boldsymbol{s}^j$, where α^j is the scaling factor of the j-th source. Although this model is commonly used to demonstrate separation abilities, anechoic noiseless environments are hard to find in the real world.

2.2. Noisy reverberant room

To simulate a more real-world setting an Acoustic Transfer Function (ATF) which relate the sources and the microphones is considered together with additive noise as follows: $\boldsymbol{x} = \sum_{j=1}^{C} \alpha^j \cdot \boldsymbol{s}^j * \boldsymbol{h}^j + \boldsymbol{n}$, where $\boldsymbol{h}^j$ is the ATF of the j-th source to the microphone, and $\boldsymbol{n}$ is a non stationary additive noise in an unknown Signal-to-Noise Ratio (SNR).

Under both cases, we focus on the fully supervised setting, in which we are provided with a training set $\mathcal{S} = \{\boldsymbol{x}_i, (\boldsymbol{s}_i^1, \cdots, \boldsymbol{s}_i^C)\}_{i=1}^m$, and our goal is learn a model that given an unseen mixture $\boldsymbol{x}$, outputs C separate channels, $\hat{\boldsymbol{s}}$, that maximize the Scale-Invariant Signal-to-Noise Ratio (SI-SNR) to the ground truth signals when considering reordering of the output channels $(\hat{\boldsymbol{s}}^{\pi(1)}, \cdots, \hat{\boldsymbol{s}}^{\pi(C)})$ for the optimal permutation π.

3. MODEL

We propose to jointly separate a varying number of sources using a single model with several separation heads and shared representations. The proposed architecture is depicted in Fig. 1.

Following the architecture proposed in [14], the mixed signal is first encoded using a stack of N 1D convolution with a kernel size of L and stride of $L/2$ followed by ReLU function. The 2D tensor output of the encoder is given by $\boldsymbol{z} \in \mathbb{R}^{N \times T'}$, where $T' = (2T/L) - 1$. Next, $\boldsymbol{z}$ is going through a chunking process. It is first divided into R overlapping chunks with chunk size of K and step size of P, denoted as $\boldsymbol{u}_r \in \mathbb{R}^{N \times K}$, where $r \in [1, \cdots, R]$. Then the 2D chunks are concatenated into a 3D embedded tensor $\boldsymbol{v} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_R] \in \mathbb{R}^{N \times K \times R}$. Next, a series of b Multiply-and-Concatenate (MULCAT) blocks, as proposed in [22], are employed to model the intra-chunk and inter-chunk dependencies.

We separate the mixture using several separation heads after each block $l \in \{1, \dots, b\}$ and output $\boldsymbol{o}_l$. The separation heads architecture is containing four experts alongside a gate. The n-th expert' expertise is to separate different number of speakers C_n, where $n \in \{1, \dots, 4\}$ and $C_n \in \{2, 3, 4, 5\}$, respectively. Note, all the experts and the gate share the same input $\boldsymbol{o}_l$. Each expert is comprised of a PReLU non-linearity with parameters initialized at 0.25, followed by 1×1 convolution with $C_n \cdot R$ kernels. The resulting tensor with a size of $N \times K \times C_n \cdot R$ is then divided into C_n tensors with size $N \times K \times R$, which are finally transformed to C_n waveforms samples by applying an overlap-and-add operation to the R chunks. The overlap between two successive frames is $L/2$.

The gating network is implemented as Convolutional Neural Network (CNN) using four convolution layers with $64, 32, 16, 8$ channels, respectively, followed by two fully connected layers. Each convolutional layer has a kernel size of 3 followed by PReLU and max-pooling with kernel size 2. The first fully connected layers have 100 PReLU neurons while the last layer outputs a distribution over the number of speakers. Unlike [22], we do not use any speaker identification loss. Note, that the same separation heads are applied after each block.

Training objective We optimize several loss functions to further improve models performance, where the main objective of each of the experts is the SI-SNR,

$$\text{SI-SNR}(\boldsymbol{s}^j, \hat{\boldsymbol{s}}^j) = 10 \log_{10} \frac{||\tilde{\boldsymbol{s}}^j||^2}{||\tilde{\boldsymbol{e}}^j||^2}, \tag{1}$$

where $\tilde{\boldsymbol{s}}^j = \frac{\langle \boldsymbol{s}^j, \hat{\boldsymbol{s}}^j \rangle \boldsymbol{s}^j}{||\boldsymbol{s}^j||^2}$ and $\tilde{\boldsymbol{e}}^j = \hat{\boldsymbol{s}}^j - \tilde{\boldsymbol{s}}^j$.

To tackle the permutation invariant problem we use the utterance level Permutation Invariant Training (uPIT) loss, as proposed in [7]:

$$L_{\text{uPIT}}(\boldsymbol{s}, \hat{\boldsymbol{s}}) = - \max_{\pi \in \Pi_{C_n}} \frac{1}{C_n} \sum_{j=1}^{C_n} \text{SI-SNR}(\boldsymbol{s}^j, \hat{\boldsymbol{s}}^{\pi(j)}), \tag{2}$$

where Π_{C_n} is the set of all possible permutations of $1, \dots, C_n$. We denote the optimal permutation π_o.

Next, to further improve optimization and reduce artifacts in the estimated signals, we include a frequency domain loss function. Similarly to [24, 25], we define the STFT loss to be the sum of the *spectral convergence (sc)* loss and the *magnitude* loss as follows,

$$\begin{aligned}
L_{\text{stft}} &= \sum_{j=1}^{C_n} L_{sc}(\boldsymbol{s}^j, \hat{\boldsymbol{s}}^{\pi_o(j)}) + L_{mag}(\boldsymbol{s}^j, \hat{\boldsymbol{s}}^{\pi_o(j)}), \\
L_{\text{sc}}(\boldsymbol{s}^j, \hat{\boldsymbol{s}}^{\pi_o(j)}) &= \frac{|||\text{STFT}(\boldsymbol{s}^j)| - |\text{STFT}(\hat{\boldsymbol{s}}^{\pi_o(j)})|||_F}{|||\text{STFT}(\boldsymbol{s}^j)|||_F}, \\
L_{\text{mag}}(\boldsymbol{s}^j, \hat{\boldsymbol{s}}^{\pi_o(j)}) &= \frac{1}{T} \| \log |\text{STFT}(\boldsymbol{s}^j)| - \log |\text{STFT}(\hat{\boldsymbol{s}}^{\pi_o(j)})| \|_1,
\end{aligned} \tag{3}$$

where $\|\cdot\|_F$ and $\|\cdot\|_1$ are the Frobenius the L_1 norms respectively. We define the multi-resolution STFT loss to be the sum of all STFT loss functions using different STFT parameters. We apply the STFT loss using different resolution with number of FFT bins $\in \{512, 1024, 2048\}$, hop sizes $\in \{50, 120, 240\}$, and lastly window lengths $\in \{240, 600, 1200\}$.

Lastly, we included a cyclic reconstruction L2 loss between the sum of the input mixture to the sum of the estimated sources. Defined as: $L_{\text{rec}} = \|\sum_{j=1}^{C_n} \hat{\boldsymbol{s}}^j - \boldsymbol{x}\|^2$. Notice, in the case of noisy and reverberant setup, we replace $\boldsymbol{x}$ by the sum of all clean input sources.

Overall, we minimize the following objective function,

$$L = L_{\text{uPIT}} + \lambda_{stft} \cdot L_{\text{stft}} + \lambda_{rec} \cdot L_{\text{rec}} + \lambda_{\text{gate}} \cdot L_g, \quad (4)$$

where L_g is the categorical cross-entropy loss used to optimize the gate branch. Note, the gate is constantly training regardless of the amount of sources. We calibrated all λ values on the validation set, and set $\lambda_{\text{rec}} = \lambda_{\text{gate}} = 1$, and $\lambda_{\text{stft}} = 0.5$.

While training, the number of speakers, C_n is randomly chosen in each mini-batch. Therefore, only the corresponding expert is trained at every mini-batch. During inference, the outputs of the expert with the highest probability are used.

Evaluation method While evaluating a separation model for a known the number of speakers is straightforward and can be done by using SI-SNR directly, it is unclear how to evaluate a separation model with an unknown number of speakers, since the predicted and target number of speakers can vary.

To mitigate that we follow the method proposed by [22]. Three cases are considered: i) the predicted and target number of speakers are the same, in this case, we simply compute the SI-SNR; ii) the predicted number of speakers is larger than the target number of speakers, here we compute the correlation between each predicted and target channels, and pick the C predicted channels with the highest correlation; iii) the predicted number of speakers is smaller than the target number of speakers. Here we also compute the correlation between the predicted and target channels, but then we duplicate the best-correlated signals to reach C number of channels.

The last case can be considered as a penalty for the model since the separation will always be flawed. In the second case, the model may produce a good separation despite predicting the wrong number of speakers.

4. DATASET

Under both clean and noisy settings, we assume all signals were sampled at 8 kHz. We set 20,000 examples for training, 5,000 samples for validation, and 3,000 samples for testing. We consider the anechoic signals as target supervision, thus under the noisy reverberant setting, we optimize the model to jointly do separation, denoising, and dereverberation.

Clean dataset For the clean dataset, we use the wsj0-2mix and wsj0-3mix mixtures as suggested in [1], while for wsj0-4mix and wsj0-5mix we follow the same mixture recipes as suggested in [22].

Table 1: Noisy reverberant data specification.

Room (m)	x	U[4,7]
	y	U[4,7]
	z	2.5
T_60 (sec)		U[0.16, 0.36]
Mic. Pos. (m)	x	$\frac{x_{\text{Room}}}{2}$+$U$[-0.2,0.2]
	y	$\frac{y_{\text{Room}}}{2}$+$U$[-0.2,0.2]
	z	1.5
# of speakers		{2/3/4/5}
Sources Pos. (°)	θ	U[0,180]
Sources Distance (m)		1.5+U[-0.2,0.2]
SNR	dB	U[0, 15]

Noisy reverberant dataset As for the noisy reverberant settings, we generate datasets for separating up to five different sources. The setup of the dataset is presented in Table 1. We synthetically generate noisy reverberant mixtures to mimic real-world recordings. The clean signals were taken from the WSJ0 corpus [26] and noise signals from the WHAM! noise dataset [27].

For each mixture, we first randomly sampled the number of speaker (between 2-5). Next, we randomly selected room dimensions, microphone positions, and different positions for the sources, as shown in Table 1. We generated a Room Impulse Response (RIR) using the rir_generator tool [28] for every speaker in the mixture, which was then convolved with the clean signal. The reverberant signals were then summed up together with an additive noise using random SNR.

5. EXPERIMENTAL RESULTS

We start by evaluating the proposed model while we assume the number of speakers in the mixture is known a-priori. Next, we move into comparing our system to several automatic-selection methods while the number of speakers in the recording is unknown. We conclude this section by analyzing the performance of the speaker classification branch. All results are reported for both clean and noisy reverberant environments. For the separation results, we report the SI-SNR improvement over the mixture, denoted as SI-SNRi.

5.1. Known number of speakers

We compared the proposed method to ConvTasNet [11], Dual-Path RNN (DPRNN) [14], and Gated model [22], for the case of a known number of speakers. The baseline methods were trained with a different model separating each number of speakers between two and five. We optimized all baseline models using the published code by the method's authors. All models were optimized until no loss improvement was observed on the validation set for five epochs using Adam optimizer with a learning rate of 3×10^{-4} and a batch size of 2.

Table 2: Performance of various models as a function of the number of speakers under the clean and noisy reverberant setups. In the following results, we assume the number of speakers in the mixture is known a-priori. All results are reported in SI-SNRi.

Model	2spk	3spk	4spk	5spk	2spk	3spk	4spk	5spk
	Clean				Noisy-reverberant			
ConvTasNet [11]	15.33	12.71	8.52	7.04	8.97	7.46	6.31	5.53
DPRNN [14]	18.81	14.68	10.39	8.72	10.24	8.34	6.92	5.89
Gated [22]	**20.12**	16.85	12.88	10.56	10.66	8.93	7.42	6.35
Ours	19.43	**17.26**	**13.93**	**11.77**	**11.48**	**10.73**	**9.48**	**8.49**

Table 2 presents the separation results. The proposed method is superior to the baseline methods by a significant margin, with one exception of two speakers in an anechoic room. These results suggest that using shared representation together with classifying the number of speaker in the mixture are beneficial specifically when considering more than two speakers or a noisy environment.

Notice, the noisy dataset is significantly more challenging than the clean dataset since the models are required to not only separate the sources but also reduce their reverberation and additive noise. Therefore all models suffer a degradation in performance compared to the clean dataset.

5.2. Unknown number of speakers

Next, we consider the case of an unknown number of speakers. We compared the proposed method to several automatic selection algorithms for the number of speakers in the recording. Specifically, we compared our model to i) [22] which trained a separate model to separate a different number of speakers, denoted as Ensemble; ii) [7, 21] which trains one model to separate the maximum number of speakers, denoted as MaxOut. We optimized the MaxOut method with and without speaker classification loss. Notice, both methods use a silent detection algorithm on top of the model's output to produce the final separation. In contrast, our work uses a speaker classification branch, we use its output to determine the number of speakers in the mixture.

For a fair comparison, all separation models are based on Gated [22], where we only change the selection algorithm. Results presented in Table 3. The proposed method is superior to the baseline methods under both clean and noisy scenarios. Notice, sharing internal representation yields in a better separation performance, while including several separation heads instead of the MaxOut method further improves the results, specifically under noisy environments. Interestingly, including the classification branch did not improve performance for the MaxOut method.

Lastly, we report the classification results obtained by our model and compared them to the silent detection algorithm as in [22]. The results are depicted in Fig. 2. Including a dedicated branch for speaker separation evidently provides a boost in classification performance, especially in noisy reverberant environments. As a side-note: we also experimented with optimizing the classification model using spectral feature rather than joint optimization with the separation heads. This, however, provided inferior performance.

Table 3: A comparison of several automatic selection algorithms for speaker separation while considering the number of speakers in the mixture to be unknown. All results are reported in SI-SNRi.

Model	2spk	3spk	4spk	5spk	2spk	3spk	4spk	5spk
	Clean				Noisy-reverberant			
Ensemble ([22])	18.63	14.62	11.48	10.37	10.24	8.59	7.07	6.21
MaxOut w/o Cls. ([7, 21])	19.29	16.8	13.34	11.31	10.59	9.41	7.92	7.5
MaxOut w/ Cls. ([7, 21])	19.11	16.71	13.35	11.29	10.58	9.39	7.97	7.51
Ours	**19.41**	**17.05**	**13.91**	**11.71**	**11.45**	**10.6**	**9.36**	**8.31**

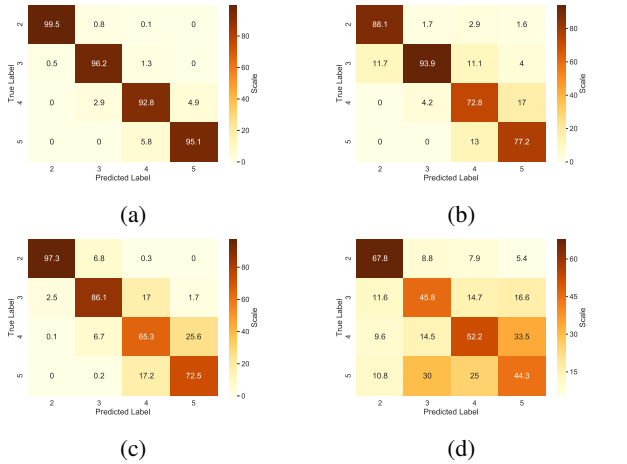

Fig. 2: Confusion matrix for model selection results using clean and noisy datasets. Results are reported for both the proposed model (Fig. 2a (clean) and Fig. 2c (noisy)) and the MaxOut model using silent detection method as proposed in [22] (Fig. 2b (clean) and Fig. 2d (noisy)). Acc. is presented inside each cell in the matrix.

It is worth mentioning that although SI-SNRi results are superior to the baseline methods while listening to the separations there still much room for improvement, especially when considering the mixtures with four or five speakers under noisy reverberant environments. Nevertheless, these separations can still be used as prior statistics for next-phase multi-channel speech processing.

6. CONCLUSIONS

Single-channel source separation is a challenging task, especially when considering a large or unknown number of speakers in noisy reverberant environments. In this work, we introduce a neural net model that handles the uncertainty regarding the number of speakers under real-world conditions. The success of our work under practical settings stems from the use of a shared representation with a multi-task loss function. Empirical results suggest the proposed method is superior to the baseline models both in terms of separation and classifying the number of speakers in the mixture.

7. REFERENCES

[1] John R Hershey et al., "Deep clustering: Discriminative embeddings for segmentation and separation," in *ICASSP*, 2016.

[2] Dong Yu et al., "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *ICASSP*, 2017.

[3] S. Gannot et al., "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 692–730, Apr. 2017.

[4] Shoji Makino, *Audio Source Separation*, Springer, 2018.

[5] Aapo Hyvärinen and Erkki Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[6] Prem Seetharaman et al., "Bootstrapping single-channel source separation via unsupervised spatial clustering on stereo mixtures," *arXiv preprint arXiv:1811.02130*, 2018.

[7] M. Kolbæk et al., "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[8] Zhuo Chen, Yi Luo, and Nima Mesgarani, "Deep attractor network for single-microphone speaker separation," in *ICASSP*, 2017.

[9] Zhong-Qiu Wang et al., "Alternative objective functions for deep clustering," in *ICASSP*, 2018.

[10] Zhong-Qiu Wang, Ke Tan, and DeLiang Wang, "Deep learning based phase reconstruction for speaker separation: A trigonometric perspective," in *ICASSP*, 2019.

[11] Yi Luo and Nima Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, pp. 1256–1266, 2019.

[12] Daniel Stoller, Sebastian Ewert, and Simon Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *arXiv preprint arXiv:1806.03185*, 2018.

[13] Shrikant Venkataramani and Paris Smaragdis, "End-to-end source separation with adaptive front-ends," *CoRR*, vol. abs/1705.02514, 2017.

[14] Yi Luo, Zhuo Chen, and Takuya Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP*, 2020.

[15] Liwen Zhang et al., "Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks," in *International Conference on Multimedia Modeling*, 2020.

[16] Neil Zeghidour and David Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *arXiv preprint arXiv:2002.08933*, 2020.

[17] Keisuke Kinoshita et al., "Listening to each speaker one by one with recurrent selective hearing networks," in *ICASSP*, 2018.

[18] Naoya Takahashi et al., "Recursive speech separation for unknown number of speakers," *arXiv preprint arXiv:1904.03065*, 2019.

[19] Jing Shi et al., "Sequence to multi-sequence learning via conditional chain mapping for mixture signals," *arXiv preprint arXiv:2006.14150*, 2020.

[20] Thilo von Neumann et al., "Multi-talker asr for an unknown number of sources: Joint training of source counting, separation and asr," *arXiv preprint arXiv:2006.02786*, 2020.

[21] Yi Luo, Zhuo Chen, and Nima Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 787–796, 2018.

[22] Eliya Nachmani, Yossi Adi, and Lior Wolf, "Voice separation with an unknown number of multiple speakers," in *ICML*, 2020.

[23] Matthew Maciejewski et al., "Whamr!: Noisy and reverberant single-channel speech separation," in *ICASSP*, 2020.

[24] Ryuichi Yamamoto et al., "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP*, 2020.

[25] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim, "Probability density distillation with generative adversarial networks for high-quality parallel waveform generation," *preprint arXiv:1904.04472*, 2019.

[26] John Garofolo, David Graff, Doug Paul, and David Pallett, "Csr-i (wsj0) complete ldc93s6a," *Web Download. Philadelphia: Linguistic Data Consortium*, vol. 83, 1993.

[27] Gordon Wichern et al., "Wham!: Extending speech separation to noisy environments," in *Interspeech*, 2019.

[28] Emanuel AP Habets, "Room impulse response generator," *Technische Universiteit Eindhoven, Tech. Rep*, vol. 2, no. 2.4, pp. 1, 2006.