Evaluating feature importance for speaker separation models

Deeksha Prabhakar*, Jose Ignacio Pozuelo*, Daniel Pluth†, Ayush Panda‡, and Vijay K. Gurbani*†
*Illinois Institute of Technology
dprabhakar@hawk.iit.edu, jpozuelosaizdebustam@hawk.iit.edu, vgurbani@iit.edu
†Vail Systems, Inc.
dpluth@vailsys.com, vgurbani@vailsys.com
‡University of Illinois
apand6@uic.edu

***Abstract*—Computational speaker separation, or multi-talker separation, attempts to use neural network models to separate a stream containing multiple speakers into an individual stream for each speaker. Such separation has advantages in many situations, for example, an emergency 911 call may contain multiple people talking simultaneously; in a call-center scenario, the agent may be presented with an audio stream that contains a customer speaking while a television is playing in the background; interviews and political debates often consist of speakers talking over each other. In all of these cases, separating the mixed stream into individual streams enables back-end processes to detect which particular stream is of importance, and provides for greater conversational intelligence from the separated streams when compared to the mixed stream. In this paper, we examine which features are important to the speaker separation problem; we comprehensively examine 14 characteristic features of a mixed audio stream to determine the subset of features that lead to a cleaner separation. Our key results are that four features, namely minimum intensity, minimum pitch, difference in intensity, and difference in pitch lead to superior speaker separation. By providing significant evidence into the relationship between input audio features and separation efficacy, this work contributes towards optimizing novel strategies for speaker separation systems.**

***Index Terms*—Speech Separation, Audio Signal Processing, Feature Extraction, Convolutional Neural Networks (CNNs), Permutation Feature Importance, SHAP values, Transformer Models, Self-Attention Mechanism**

I. Introduction and problem statement

Many real-world communication scenarios require understanding speech streams that are intertwined. For example, emergency 911 call centers routinely get calls where the caller's voice is interspersed with other surrounding voices (background conversations, the television, or another person arguing with the caller). Similarly, in a political debate, or an interview, there may be two people speaking simultaneously such that the resulting conversation is unintelligible. In many call center operations, the agent handles calls from customers who are in a noisy environment with the mic picking up nearby conversations. In all of these cases, separating the mixed stream into individual streams enables back-end processes to identify the most important stream for further processing. Further, decomposing the mixed stream into its individual streams allows for greater conversational intelligence from the separated streams when compared to the mixed stream.

We note that the speaker separation problem is a subset of speech separation but distinct from speaker diarization. Speech separation can be considered as a superset of speaker separation as the background interference from which speech is separated can consist of background noise, music, and even other speakers [18]. However, speaker diarization is concerned with labeling portions of audio with a speaker identity to identify "who spoke when" [16].

Single-channel, or monaural speaker separation is a fundamental problem in audio and speech processing with early work on separation based primarily on signal processing at the spectrum level of the input signal [5], [11]. Recent work in the area of speaker separation is based on deep learning neural networks [7], [8], [13], [17], [23], [24]. In such approaches, a deep neural network learns to predict time-frequency masks of two speakers[1] in a mixture consisting of a monaural source. However, the disadvantage of using deep neural networks is their lack of interpretability; it is not entirely clear what is the cause when a neural network models excels at separation. Are certain features of the mixed audio stream more important than others, and if so, what is the subset of those features? Further, it remains unclear whether the high efficacy demonstrated on the held-out test dataset will translate to an equivalently high efficacy for out-of-distribution samples.

Contributions: This work systematically answers these questions with an ultimate goal of informing future speaker separation systems of novel strategies to explore for superior separation. Concretely, our contributions are:

- We enumerate a set of features that leads to superior optimal separation,
- We demonstrate that these features are invariant of the dataset from which they are collected, i.e., they are generalizable across multiple out-of-distribution datasets,
- We make publicly available two datasets collectively consisting of 1,400 individual audio files (9,803 seconds) and 700 mixed audio files for further research into dependable speaker separation. The datasets are available at https://github.com/vkgurbani/ieee_southeastcon_2024.

[1]We limit our investigation to an monaural mixed stream consisting of two speakers, as the current state of art open source models are not competitive for separating mixes containing more than two speakers [14].

The rest of the paper is organized as follows. Section II puts our work in the context of other literature evaluating speaker separation systems. Section III discusses the datasets that we used in our approach, and Section IV details our overall methodology on evaluating the efficacy of a subset of features that contribute to superior separation. Section V subsequently outlines our feature importance process and presents a set of features that optimize speaker separation; Section VI demonstrates the generalizability of the chosen feature set on a separate, out-of-distribution dataset. Finally, we conclude and outline future work in this area in Section VII.

II. Related work

In placing our work in the context of existing literature, we focus on speaker separation rather than the more general area of speech separation; and within speaker separation, we review literature that *evaluates* speaker separation systems rather than work that *describes* novel speaker separation algorithms.

Much of the literature in evaluating speaker separation systems uses signal-based metrics such as the signal-to-noise (SNR) ratio [4], [6], [20], [21]; such metrics have a weak correlation with ASR accuracy. Our work, takes a much different approach. First we enumerate the features of the mixed streams to determine which particular features are important, and second, we evaluate the speaker separation system through a word error rate (WER[2]) metric obtained from the post-separation stream by sending it through an ASR and comparing the resulting WER with the WER of the stream before it was mixed.

Wichern et al. [22] evaluate WSJ0-2mix dataset in varying outdoor environments to evaluate robustness to noise; they do not consider ASR evaluation of the separated streams or feature importance in their work. Bahmaninezhad et a;. [1] simulated a multi-channel spatialized reverberate dataset and evaluated the separation using signal-based metrics as well as WER; however, their ASR system was trained on clean non-reverberate data thus rendering high WER rates on the separated data streams (an average WER of 44% across the models in their experiment).

Other works have used exogenous information to separate speakers; Wang et al. [19] use "speaker inventories" to distinguish speakers. A speaker inventory consists of a list of speaker profiles collected from the speakers. It is not entirely clear that such inventories will be available in for different scenarios where speaker separation may be used. For example, in an emergency 911 use case or a call-center conversation, such inventories will not be readily available. Khan et al. [9] utilize visual speech features by focusing on the mouth region of each speaker. This will require a camera to capture the video of the speakers, a constraint for most use cases where only the audio is available for analysis.

[2]The WER is a widely accepted metric to measure the performance of a speech recognition system. We define it formally in Section IV.

TABLE I: Summary of derived datasets

	Number of files	Number of seconds (Min/1st. Qu./Median/Mean/3rd Qu./Max)
Dataset-1 VoxCeleb	400	2,866 (4.0 / 5.4 / 7.0 / 7.2 / 8.8 / 11.0)
Dataset-2 Mozilla Common Voice	1,000	6,936 (3.0 / 5.0 / 7.0 / 7.0 / 9.0 / 10.9)

III. Datasets

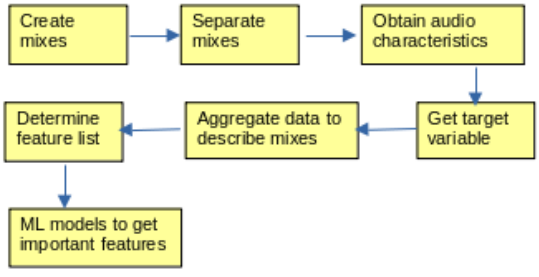
We used two datasets for our work: the VoxCeleb dataset [15] and the Mozilla Common Voice dataset (Common Voice Corpus 12.0[3]). The VoxCeleb dataset is a collection of more than 100,000 utterances attributed to 1,251 distinct celebrities, sourced from videos hosted on the YouTube platform. Notably, the dataset encompasses speech samples from a heterogeneous spectrum of speakers, representing diverse demographics in terms of age, profession, accent, and ethnicity. An inherent characteristic of VoxCeleb is the incorporation of "in the wild" speech recordings, thereby encapsulating a plethora of real-world conditions including ambient noise, instances of laughter, speech overlap, and variable vocal demeanor. The Mozilla Common Voice dataset consists of MP3 audio file and an associated text file. A distinguishing feature of the dataset is its incorporation of 3,161 recorded hours, within which a significant portion is enriched with supplementary demographic attributes, including variables such as age, gender, and accent.

We created two derivative datasets from the VoxCeleb and Mozilla Common Voice datasets. The first dataset ("Dataset-1") was derived from the VoxCeleb dataset and was used to derive the important features that lead to a clean separation (cf. Section V). The second dataset ("Dataset-2") was curated from Mozilla Common Voice. Table I summarizes the datasets.

IV. Methodology

We now describe the experimental framework for conducting a systematic investigation into understanding the feature importance for speaker separation. The overall process is depicted in Figure 1.

Fig. 1: Methodological steps



The mixes were created from VoxCeleb and Mozilla Common Voice as described in the previous section. From VoxCeleb dataset we randomly chose 400 individual streams and

[3]https://commonvoice.mozilla.org/en/datasets

from the Mozilla Common Voice corpus we randomly chose 1,000 streams. Streams were chosen from these corpora such that the mean of the streams was around 7s with a standard deviation of 2s; we chose these values because shorter audio streams may not offer enough context to the deep learning separation model to allow effective separation, while longer audio will simply require more processing without affecting the accuracy. Each stream was normalized to achieve a uniform volume level. Once the streams were normalized, we mixed a pair of streams; we do not cross-contaminate the mixes, i.e., streams from VoxCeleb dataset were only mixed with another stream from the same dataset (and similarly for Mozilla Common Voice). The mixing resulted in 200 mixed streams for VoxCeleb and 500 mixed streams for Mozilla Common Voice. The mixed streams from the VoxCeleb dataset (Dataset-1) are used in Section V to gather the important features, whilst the mixed streams from the Mozilla Common Voice dataset are used in Section VI to demonstrate the generalizability of the selected features on an out-of-distribution corpus.

Formally, let S be a set of audio streams $\{s^1, s^2, ..., s^n\}$ with $|S| = n$. Let $(s^i, s^{i+1}) \in S$, $i \in \{1, n\}$ be two adjacent audio streams that need to be mixed. The mixture procedure can be abstracted mathematically as a transform T shown in Equation 2 that maps adjacent pair of elements in S to an element in M:

$$S \xrightarrow{\mathrm{T}} M \tag{1}$$

More specifically, for each pair of adjacent streams in S, transform T mixes these into a single audio stream in M, or

$$\begin{aligned} &(s^i, s^{i+1}) \xrightarrow{\mathrm{T}} M_k, \\ &\{i \in \{1, n-1\} \mid i \text{ is odd}\}, k \in \{1, n/2\}, \end{aligned} \tag{2}$$

Note that $|M| = n/2$.

Next, each mixed audio stream in M are separated using a transform, T^{-1}, that decomposes each element of M to its corresponding pair of streams in S', i.e.,

$$M \xrightarrow{\mathrm{T}^{-1}} S' \tag{3}$$

Mathematically T^{-1} transform is the inverse operation of Equation 2, i.e., every mixed stream in M is decomposed into its individual streams $(d^i, d^{i+1}) \in S'$ through the inverse transform T^{-1}:

$$M_k \xrightarrow{\mathrm{T}^{-1}} (d^{(k*2)-1}, d^{(k*2)}) \; k \in \{1, n/2\} \tag{4}$$

Note that $|S'| = |S| = n$. Practically, T could be realized as the Sound eXchange (SoX[4]) audio editing software, and the SepFormer deep neural speech separation model [17] can be used as the inverse, T^{-1}.

Given an s^i and its corresponding d^i, the next step would be to measure the degradation of the audio signal between the original stream (s^i) and the corresponding decomposed stream (d^i). To measure this degradation, we use the Amazon Transcribe Automatic Speech Recognition (ASR) service[5]. This service accepts an audio file and produces a text transcript corresponding to the audio file. We uploaded the audio files from S and from S' to the platform and received their text transcripts. Recall that S contains the individual audio streams before they were mixed, and S' contains the audio streams that resulted from separating the mixtures.

Clearly, because the mixing process will degrade the signal to some extent, a metric is required to measure this degradation; we choose the Word Error Rate (WER) as this metric [10]. The WER is a widely accepted metric to measure the performance of a speech recognition system. It is calculated as the number of substitutions, deletions, and insertions required to transform a *hypothesis* string to a *reference* string, or

$$\text{WER} = \frac{\text{substitutions+deletions+insertions}}{\text{word count}} \tag{5}$$

The reference string acts as the "ground truth", and the number of edits are calculated to get the hypothesis string to match the reference string. The WER should be minimized, i.e., smaller values are preferred. In practice, a WER > 0.15 prohibits further processing as it represents enough noise in the transcript to prohibit downstream computations that use the text transcript.

In order to have high confidence in the results of this experiment, we used human transcriptions to create the ground truth. That is, for the subset of $s^i \in S$ that corresponded to the observations in Dataset-1, each observation in this subset was heard by a human and transcribed. For the remaining observations in S that corresponded to Dataset-2, we chose those audio streams from Mozilla Common Voice that had human transcripts. Mozilla Common Voice has a community process by which certain audio streams also have the corresponding transcript as meta-data; when choosing the audio streams from Mozilla Common Voice, we only chose those that had valid transcripts. Armed with these human-generated transcripts, we used Amazon Transcribe and Google Speech-to-text[6] ASR to generate WER for all observations in S. For our dataset, Amazon Transcribe yielded a lower WER; for the rest of our experiments, we use Amazon Transcribe.

With the sets S, M, and S' ready, we now proceed to deriving the features themselves, and to understand their importance through training machine learning classifiers (Section V). Once the important features have been determined, we demonstrate their generalizability by studying their behaviour on a completely different, out-of-distribution dataset to determine whether they remain invariant (Section VI).

V. Deriving feature importance

To understand which features lead to better separation, we first had to define the features. The work of defining features was done on Dataset-1, i.e., the 400 observations (leading to 200 mixes) of the VoxCeleb dataset.

[4]SoX Sound eXchange, http://sox.sourceforge.net/.

[5]https://aws.amazon.com/pm/transcribe/

[6]https://cloud.google.com/speech-to-text

A. Feature Selection

In order to identify important audio characteristics for the separation of audios, a set of candidate features was generated. These features were generated with no assumed knowledge. A subset of the audios of S was examined manually to determine simply what features varied between files in the set. The data for each of the features was generated algorithmically using Praat [2] where possible and by careful listening otherwise. Once collected, these features were further pruned according to variance that existed within the dataset. If the vast majority of the data had no variance for a particular feature, it was excluded from the analysis. This left us with a set of seven primary features f_S:

1) Perceived gender
2) Presence of two or more speakers
3) Presence of background noise
4) Minimum pitch
5) Maximum pitch
6) Minimum intensity
7) Minimum intensity

When the mixtures M were created, a new set of mixed stream audio features f_M were created based on f_S. Features that were factors were appropriately one-hot encoded, whereas numerical values were maintained. This resulted in the following feature list f_M:

1) Gender: either male-male, male-female, or female-female pairs.
2) Multiple speakers: not present, present in one stream, present in both streams.
3) Background noise: 19 classes based on noise combinations as well as an additional class for noise absent.
4) Minimum pitch: The overall lowest pitch value.
5) Minimum intensity: The overall lowest volume value.
6) Pitch difference: The range between the highest pitch and the lowest pitch.
7) Intensity difference: The range between the highest volume and the lowest volume.

As an example, consider two adjacent streams $(s^i, s^{(i+1)}) \in S$ as shown in Table II. From these adjacent streams, a single observation is created that contains the feature list f_M shown in Table III. The values in Table III are derived as follows: Gender is encoded into a categorical (integer) value, for example if both streams in Table II have M/M, the value is 0, if M/F, value is 1, etc., and similar categorical values are derived for multiple speakers. Background noise is categorized with discrete integer values corresponding to the type of background noise. Pitch min and intensity min are simply the minimum values of the corresponding pair of rows in Table II, while pitch difference and intensity difference are calculated as follows:

$$\begin{aligned}&\text{pitch diff} = max(\text{pitch}(s^i), \text{pitch}(s^{(i+1)})) - \\ &min(\text{pitch}(s^i), \text{pitch}(s^{(i+1)})) \\ &\text{intensity diff} = max(\text{intensity}(s^i), \text{intensity}(s^{(i+1)})) - \\ &min(\text{intensity}(s^i), \text{intensity}(s^{(i+1)}))\end{aligned} \tag{6}$$

We create a feature vector matrix with predictors shown in Table III for all observations in S.

B. Target Variable Encoding

With the predictors defined as described in the previous section, we now proceed to create a target variable. We derive a binary target using the WER; the WER is calculated by comparing transcripts generated from $d^i \in S'$ compared to the 'ground truth' of $s^i \in S$ (here, d^i is the hypothesis string, while s^i is the reference string). We note the WER of the pair of streams before they are mixed and again after they are separated. The difference between the WER is thus calculated, and if the difference is ≤ 0.15, we assign 1 as a target variable, else we assign 0.

As an example, consider again the pair of streams in Table II; assume that the WER before mixing and after separation is shown in Table IV. The average WER, $\bar{W}$, is calculated as follows:

$$\bar{W} = \frac{1}{2} * ((d^i - s^i) + (d^{(i+1)} - s^{(i+1)})) \tag{7}$$

where $s^i \in S$ and $d^i \in S'$, and $s^{(i+1)} \in S$ and $d^{(i+1)} \in S'$ represent the WER of the respective stream before mixing and after separation. The average difference in WER represents the loss of audio signal between the original streams $s^i, s^{(i+1)}) \in S$ and their respective decomposed streams in $d^i, d^{(i+1)}) \in S'$. The target variable is then assigned as follows:

$$\text{target label} = \begin{cases} 1 & \text{if } \bar{W} \leq 0.15 \\ 0 & \text{if } \bar{W} > 0.15 \end{cases} \tag{8}$$

At the end of this process, each pair of adjacent streams in S have been summarized into one observation as shown in Table III, and using S', a target variable has been assigned as shown in Equations 7 and 8. We now proceed to use this feature vector matrix to determine feature importance.

C. Feature importance

To determine which features are most influential in enabling high quality speech separation, we trained several models to predict WER using our selected features. The importance of the features for the models to make the prediction informs us of their importance in the speech separation model.

The following models were trained:

1) Logistic Regression
2) Decision Trees
3) Support Vector Machine (SVM)
4) XGBoost
5) AdaBoost

There are several ways to extract feature importance from models, but it was important that the method be model agnostic

TABLE II: Example adjacent stream features in f_S

Audio file	Gender	2 or more speakers?	Background noise	Pitch max	Pitch min	Intensity max	Intensity min
stream1.wav	M	No	Music	527.22	90.57	78.57	22.96
stream2.wav	F	Yes	None	597.80	70.02	78.76	31.74

TABLE III: Observation with features f_M

Gender	2 or more speakers?	Background noise	Pitch min	Intensity min	Pitch diff.	Intensity diff.
1	1	15	70.02	22.96	527.78	55.80

TABLE IV: Calculating target variable

	WER before mixing	WER after separation
stream1.wav	0.10	0.20
stream2.wav	0.17	0.24

so that the results could be easily compared. Common methods like Gini impurity are not applicable to non-tree based models. Ultimately two techniques were chosen, Permutation importance and SHapley Additive exPlanations (SHAP[7]) values [3], [12].

Permutation importance works by shuffling one feature at a time and evaluating the subsequent loss in model accuracy to determine how valuable that information is for making correct predictions. The psuedocode in Algorithm 1 shows the general method for computation.

Algorithm 1: Permutation importance algorithm

function Permutation importance (m, D);
Input : predictive model m trained on dataset D
Output: importance i_f of each feature f of D
Compute reference accuracy s of m on D
for *f of D* **do**
  for *k in 1,...,K* **do**
    Shuffle column f to generate $\tilde{D}_{f,k}$
    Compute score $s_{f,k}$ of m on $\tilde{D}_{f,k}$
  end
end
Return:
$i_f = s - \frac{1}{K}\sum_{1=1}^{K} s_{k,f}$

SHAP values are used to explain why a particular data point resulted in the model producing a particular prediction. It uses a game theory approach to measure each feature's contribution to a final outcome. The method involves constructing many variants of a given model, but with different subsets of the available features. In this way, the overall contribution of a single variable can be understood while controlling for the other features in the set.

Figure 2 shows a graphical representation of the total contributions of each feature for each datapoint to result in their corresponding WER prediction. To predict feature importance, the sum is taken of the feature magnitude over all samples in the dataset. This tells us which variables that contributed the most to predictions for the samples in our set.

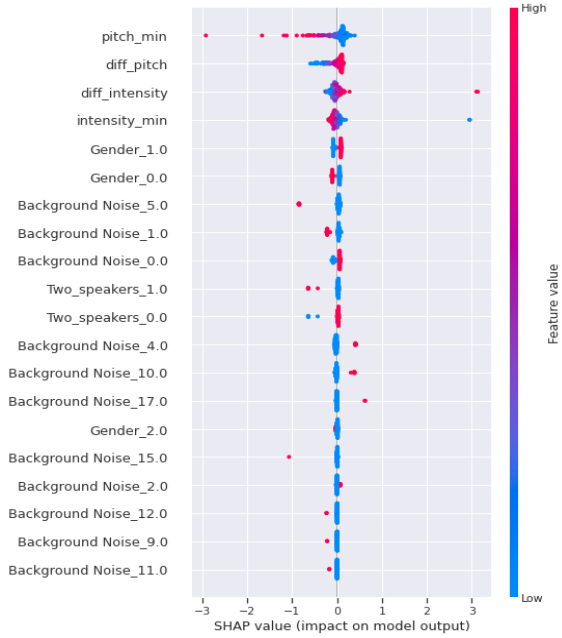


Fig. 2: SHAP Values

For the two importance calculation techniques, we generated scores for each feature in the dataset from each of the five models tested. The scores were then aggregated across the models to provide rankings per technique. The aggregation was scaled according to each model's accuracy of prediction on the dataset. Table V shows the rankings for the top four variables on the Voxceleb dataset.

TABLE V: VoxCeleb Results

Rank	Permutation Importance	SHAP Results
1	diff_intensity	diff_intensity
2	intensity_min	pitch_min
3	diff_pitch	intensity_min
4	pitch_min	diff_pitch

VI. Generalizablitity of Chosen Features

In order to understand to what degree the results are generalizable, the experiment is performed on both the VoxCeleb and

[7]SHAP is a game theoretic approach to measure how much each feature contributes to the output of a machine learning model.

Mozilla CommonVoice dataset. Since the majority of Voxceleb is recorded via interviews, the recording equipment tends to be high quality and the environment varies between live audience and clean recording rooms. Mozilla CommonVoice however, is user submitted by volunteers. The recording equipment likely varies widely, with laptop microphones likely making up a majority of the equipment. Likewise, the environment of the recordings may be noisy or relatively clean, but is unlikely to be studio recording conditions.

The experiment was repeated with Mozilla Common Voice, features extracted, models trained, feature importance calculated. The resulting ranking of the top four variables in Table VI shows the same four variables as Voxceleb. The ranking is different, but suggests that the separation model is most sensitive to these variables and that the results hold true across distinct datasets.

TABLE VI: Mozilla CV Results

Rank	Permutation Importance	SHAP Results
1	intensity_min	intensity_min
2	pitch_min	pitch_min
3	diff_pitch	diff_pitch
4	diff_intensity	diff_intensity

VII. Conclusion and future work

In summary, this study meticulously explored feature-based speech separation, leveraging pre-trained deep neural networks. Utilizing the VoxCeleb corpus and the SepFormer model features such as intensity and pitch consistently emerged as influential markers for successful separation across various supervised learning models. The application of permutation importance and SHAP values quantified their significance, affirming their pivotal role in achieving high-quality speaker separation and reinforcing their robustness across different datasets, as validated by experiments on the Mozilla CommonVoice dataset.

In conclusion, the study provides substantial evidence regarding the crucial features governing speaker separation outcomes, emphasizing the potential impact of strategic feature selection on separation efficacy. These findings lay a foundation for further advancements in audio source separation, with implications for improving real-world applications such as emergency call processing, call-center scenarios, and political debates where multiple speakers may speak simultaneously. The identified features' robustness and generalizability underscore their relevance and promise in the continued evolution of speaker separation technologies. Looking forward, future research endeavors in perceptual analysis, automation, dataset diversity, and algorithmic innovation aim to push the boundaries of audio source separation, contributing to the development of more sophisticated and efficient techniques.

References

[1] F. Bahmaninezhad et al. A comprehensive study of speech separation: spectrogram vs waveform separation. *arXiv preprint arXiv:1905.07497*, 2019.

[2] P. Boersma and D. Weenink. Praat, a system for doing phonetics by computer. *Glot international*, 5:341–345, 01 2001.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] Z. Chen et al. Continuous speech separation: Dataset and analysis. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7284–7288, 2020.

[5] S. Choi et al. Blind source separation and independent component analysis: A review. *Neural Information Processing-Letters and Reviews*, 6(1):1–57, 2005.

[6] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent. Librimix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*, 2020.

[7] J. Du et al. Speech separation of a target speaker based on deep neural networks. In *2014 12th International Conference on Signal Processing (ICSP)*, pages 473–477. IEEE, 2014.

[8] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Deep learning for monaural speech separation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1562–1566. IEEE, 2014.

[9] F. U. Khan, B. P. Milner, and T. Le Cornu. Using visual speech information in masking methods for audio speaker separation. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 26(10):1742–1754, oct 2018.

[10] D. Klakow and J. Peters. Testing the correlation of word error rate and perplexity. *Speech Communication*, 38(1):19–28, 2002.

[11] G. Logeshwari and G. Anandha Mala. A survey on single channel speech separation. In *Advances in Communication, Network, and Computing: 3rd International Conference, February 24-25, 2012, Revised Selected Papers 3*, pages 387–393. Springer, 2012.

[12] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774. 2017.

[13] Y. Luo, Z. Chen, and T. Yoshioka. Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 46–50. IEEE, 2020.

[14] E. Nachmani et al. Voice separation with an unknown number of multiple speakers. In *Proceedings of the 37th Int'l. Conf. on Machine Learning*, volume 119, pages 7164–7175. PMLR, 13–18 Jul 2020.

[15] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[16] T. J. Park et al. A review of speaker diarization: Recent advances with deep learning. *Computer Speech Language*, 72:101317, 2022.

[17] C. Subakan et al. Attention is all you need in speech separation. In *ICASSP 2021 - 2021 IEEE Int'l. Conf. on Acoustics, Speech and Signal Processing*, pages 21–25, 2021.

[18] D. Wang and J. Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1702–1726, 2018.

[19] P. Wang, Z. Chen, D. Wang, J. Li, and Y. Gong. Speaker separation using speaker inventories and estimated speech. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:537–546, dec 2020.

[20] Z.-Q. Wang et al. Deep learning based phase reconstruction for speaker separation: A trigonometric perspective. In *ICASSP 2019 - IEEE Int'l. Conf. on Acoustics, Speech and Signal Processing*, pages 71–75, 2019.

[21] Z.-Q. Wang and D. Wang. Combining spectral and spatial features for deep learning based blind speaker separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(2):457–468, 2019.

[22] G. Wichern et al. Wham!: Extending speech separation to noisy environments. In *Proc. Interspeech*, Sept. 2019.

[23] X.-L. Zhang and D. Wang. A deep ensemble learning method for monaural speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 24(5):967–977, 2016.

[24] Z. Zhang, Y. Xu, M. Yu, S.-X. Zhang, L. Chen, and D. Yu. Adl-mvdr: All deep learning mvdr beamformer for target speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6089–6093. IEEE, 2021.