Speaker-Independent Speech Separation With Deep Attractor Network

Yi Luo, Zhuo Chen, and Nima Mesgarani

***Abstract*—Despite the recent success of deep learning for many speech processing tasks, single-microphone, speaker-independent speech separation remains challenging for two main reasons. The first reason is the arbitrary order of the target and masker speakers in the mixture (permutation problem), and the second is the unknown number of speakers in the mixture (output dimension problem). We propose a novel deep learning framework for speech separation that addresses both of these issues. We use a neural network to project the time-frequency representation of the mixture signal into a high-dimensional embedding space. A reference point (attractor) is created in the embedding space to represent each speaker which is defined as the centroid of the speaker in the embedding space. The time-frequency embeddings of each speaker are then forced to cluster around the corresponding attractor point which is used to determine the time-frequency assignment of the speaker. We propose three methods for finding the attractors for each source in the embedding space and compare their advantages and limitations. The objective function for the network is standard signal reconstruction error which enables end-to-end operation during both training and test phases. We evaluated our system using the Wall Street Journal dataset (WSJ0) on two and three speaker mixtures and report comparable or better performance than other state-of-the-art deep learning methods for speech separation.**

***Index Terms*—Source separation, multi-talker, deep clustering, attractor network.**

I. Introduction

LISTENING to an individual in crowded situations often takes place in the presence of interfering speakers. Such situations require the ability to separate the voice of a particular speaker from the mixed audio signal of others. Several proposed systems have shown significant performance improvement on the separation task when prior information of speakers in a mixture is given [1], [2]. This however is still challenging when no prior information about the speakers is available, a problem known as speaker-independent speech separation. Humans are particularly adept at this task, even in the absence of any spatial separation between speakers [3]–[5]. This effortless task for humans, however, has proven difficult to model and emulate algorithmically. Nevertheless, it is a challenge that must be solved in order to achieve robust performance in speech processing tasks. For example, while the performance of current automatic speech recognition (ASR) systems has reached that of humans in clean conditions [6], these systems are still unable to perform well in noisy and crowded environments, lacking robustness when interfering speakers are present. This becomes even more challenging when separating all sources in a mixture is required, such as in meeting transcription and music separation. When signals from multiple microphones are available, beamforming algorithms can be used to improve the target-to-masker ratio [7], [8]; when only one microphone is available, however, the general problem of audio separation remains largely unresolved.

Prior to the emergence of deep learning, three main categories of algorithms were proposed to solve the speech separation problem: *statistical* methods, *clustering* methods, and *factorization* methods, with focus on different target tasks. In *statistical* methods, the target speech signal is modeled with probability distributions such as complex Gaussian [9] or methods such as independent component analysis (ICA) [10], where the interference signal is assumed to be statistically independent from the target speech. Maximum likelihood estimation method is typically applied based on the known statistical distributions of the target. In *clustering* methods, the characteristics of the target speaker, such as pitch and signal continuity are estimated from the observation and used to separate the target signal from other sources in the mixture. Methods such as computational auditory scene analysis (CASA) [11], [12] and spectral clustering [13] fall into this category [14]. *Factorization* models, such as non-negative matrix factorization (NMF) [15]–[17], formulate the separation problem as a matrix factorization problem in which the time-frequency (T-F) representation of the mixture is factorized into a combination of basis signals and activations. The activations learned for each basis signal are then used to reconstruct the target sources.

In recent years, deep learning has made important progress in audio source separation. Specifically, neural networks have been successfully applied in speech enhancement and separation [18]–[24] and music separation [25], [26] with significantly better performance than that of traditional methods. A typical paradigm for neural networks is to directly estimate T-F masks of the sources given the T-F representation of the audio mixture (such as noisy speech or multiple speakers) [22], [23], [27], [28]. This formulates the separation as a supervised single-class or multi-class regression problem. Different types of masks and

Manuscript received June 20, 2017; revised November 25, 2017 and January 12, 2018; accepted January 13, 2018. Date of publication January 23, 2018; date of current version February 8, 2018. Y. Luo, Z. Chen, and N. Mesgarani contributed equally to this work. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Simon Doclo. *(Corresponding Author: Nima Mesgarani.)*

The authors are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: yl3364@columbia.edu; zc2204@columbia.edu; nima@ee.columbia.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2018.2795749

objective functions have been proposed. For instance, phase-aware masks for enhancement and separation have been studied in [21], [29], [30].

Limitations of the previous neural networks become evident when one considers the problem of separating two simultaneous speakers with no prior knowledge of the speakers (speaker-independent scenario). Two main challenges in this situation are the so-called *permutation* problem and the *output dimension mismatch* problem. *Permutation* problem [27] refers to the observation that the order of the speakers in the target may not be the same as the order of the speakers in the output of the network. For example, when designing the target output by separating speakers S_1 and S_2, both (S_1, S_2) and (S_2, S_1) are acceptable permutations of the speakers. Once the target permutation is fixed, however, the output of the network must follow the permutation of the target. In situations where the separation is successful but the outputs have incorrect permutation compared with the targets, the output error will be large, causing the network to diverge from the correct solution. Aside from this issue, the *output dimension mismatch* problem is also a notable problem. Since the number of speakers in a mixture can vary, a neural network with a fixed number of output targets does not have the flexibility to separate the mixtures where the number of speakers is not equal to the number of output targets.

Two deep learning methods, deep clustering (DPCL) [27] and permutation invariant training (PIT) [28], have been proposed recently to resolve these problems. In deep clustering, a network is trained to generate a discriminative embedding for each T-F bin so that the embeddings of the T-F bins that belong to the same speaker are closer to each other. Because deep clustering uses the Frobenius norm between the affinity matrix of embeddings and the affinity matrix of ideal speaker assignment (e.g. ideal binary mask) as the training objective, it solves the permutation problem due to the permutation-invariant property of affinity matrices. The mask estimation process is done by applying clustering algorithms such as K-means [31] or spectral clustering [32] to the embeddings, while the assignment of the embeddings to the clusters forms the final estimation. Hence, the number of outputs is only determined by the number of target clusters. While DPCL is able to solve both the permutation and output dimension mismatch problems and produce a state-of-the-art performance, it is unable to use reconstruction error as the target for optimization. This is because the mask generation is done through a post-clustering step on the embeddings, which is done separately from the network network. In more recent DPCL work, minimizing the separation error is processed with an unfolded soft clustering subsystem for direct mask generation, and an additional mask enhancement network is followed for better performance [33]. The PIT algorithm solves the permutation problem by first calculating the training objective loss for all possible permutations for C mixing sources ($C!$ permutations), and then using the permutation with the lowest error to update the network. It solves the output dimension problem by assuming a maximum number of sources in the mixture and using null output targets (very low energy Gaussian noises) as auxiliary targets when the actual number of sources in the mixture is smaller than the number of outputs in the network. PIT was proposed in [28] in a frame-wise fashion and was later shown to have comparable performance to DPCL [34] with a deep LSTM network structure.

We address the general source separation problem with a novel deep learning framework which we call the 'attractor network'. The term "attractor" refers to the well-studied effects in human speech perception which suggest that biological neural networks create perceptual attractors (magnets). These attractors warp the acoustic feature space to draw in the sounds that are close to them, a phenomenon that is called the Perceptual Magnet Effect [35]–[37]. Our proposed model works on the same principle as DPCL by first generating a high-dimensional embedding for each T-F bin. We then form a reference point (attractor) for each source in the embedding space that pulls all the T-F bins belonging to that source toward itself. This results in the separation of sources in the embedding space. A mask is estimated for each source in the mixture using the similarity between the embeddings and each attractor. Since the correct permutation of the masks is directly related to the permutation of the attractors, our network can potentially be extended to an arbitrary number of sources without the permutation problem once the order of attractors is established. Moreover, with a set of auxiliary points in the embedding space, known as the anchor points, our framework can directly estimate the masks for each source without needing a post-clustering step as in [27] or a clustering subnetwork as in [33]. This aspect creates a system that directly generates the reconstructed spectrograms of the sources in both training and test phases.

The rest of the paper is organized as follows. In Section II, we introduce the general problem of source separation and the embedding learning method. In Section III, we describe the original deep attractor network proposed in [24]. In Section IV, we propose several variants and extensions to the original deep attractor network for better performance. In Section V, we evaluate the performance of the attractor network and analyze the properties of the embedding space.

II. Source Separation and Embedding Learning

We start with defining the general problem of single-channel speech separation, and describe how the method of embedding learning can be used to solve this problem.

A. Single-channel Speech Separation

The problem of single-channel speech separation is defined as estimating all the C speaker sources $s_1(t), \ldots, s_c(t)$ given the mixture waveform signal $x(t)$

$$x(t) = \sum_{i=1}^{C} s_i(t) \tag{1}$$

In time-frequency (T-F) domain, the complex short-time Fourier transform (STFT) spectrogram of the mixture, $\mathcal{X}(f,t)$ equals to the sum of the complex STFT spectrograms of all the sources

$$\mathcal{X}(f,t) = \sum_{i=1}^{C} \mathcal{S}_i(f,t) \tag{2}$$

Many speech separation systems use the real-valued magnitude spectrogram as the input and estimate a set of time-frequency masks for the sources. We denote the flattened magnitude spectrogram, $|\mathcal{X}(f,t)|$, as a feature vector $\mathbf{x} \in \mathbb{R}^{1\times FT}$, where F is the number of frequency channels and T is the total duration of the utterance. The flattened magnitude spectrograms, $|\mathcal{S}_i(f,t)|$, and corresponding time-frequency masks for source $i = 1, 2, \ldots, C$ are the vectors $\mathbf{s}_i \in \mathbb{R}^{1\times FT}$ and $\mathbf{m}_i \in \mathbb{R}^{1\times FT}$ respectively. The estimated magnitude spectrograms of the source i is denoted by $\hat{\mathbf{s}}_i \in \mathbb{R}^{1\times FT}$, and calculated by

$$\hat{\mathbf{s}}_i = \mathbf{x} \odot \mathbf{m}_i \tag{3}$$

subject to

$$\sum_{i=1}^{C} \mathbf{m}_i = \mathbf{1} \tag{4}$$

where $\odot$ is element-wise multiplication and $\mathbf{1} \in \mathbb{R}^{1\times FT}$ denotes all-one vector. Commonly used masks include ideal binary mask (IBM) [38], ideal ratio mask (IRM) [39], and 'wiener-filter' like mask (WFM) [40]:

$$\begin{cases} IBM_{i,ft} = \delta(|\mathbf{s}_{i,ft}| > |\mathbf{s}_{j,ft}|), \quad \forall j \neq i \\ IRM_{i,ft} = \frac{|\mathbf{s}_{i,ft}|}{\sum_{j=1}^{C} |\mathbf{s}_{j,ft}|} \\ WFM_{i,ft} = \frac{|\mathbf{s}_{i,ft}|^2}{\sum_{j=1}^{C} |\mathbf{s}_{j,ft}|^2} \end{cases} \tag{5}$$

where $\delta(x) = 1$ if expression x is true and $\delta(x) = 0$ otherwise. The reconstruction of the time-domain signals is done by calculating the inverse short-time Fourier transform using the estimated magnitude spectrograms $\hat{\mathbf{S}}_i$ and the phase of the mixture spectrogram, $\angle\mathcal{X}(f,t)$.

B. Source Separation in Embedding Space

High-dimensional embedding is a powerful and commonly used method in many tasks such as natural language processing and manifold learning [41]–[44]. This technique maps the signal into a high-dimensional space where the resulting representation has desired properties. For example, word embedding is currently one of the standard tools to extract the relationship and connection between different words, and serves as a front-end to more complex tasks such as machine translation and dialogue systems [45], [46]. In the problem of source separation in T-F domain, a high-dimensional embedding for each T-F bin is found and speech separation is formulated as a source segmentation problem in the embedding space [12], [13].

The embeddings can be either knowledge-based or data-driven. CASA is a popular frameworks for using specially designed features to represent the sources [47], where different types of acoustic features are concatenated to produce a high-dimensional embedding which represents the sound source in different time-frequency coordinates. An example of the data driven embedding approach for speech separation is the recently proposed deep clustering method (DPCL) [27]. DPCL uses a neural network model to learn embeddings of T-F bins such that to minimize the in-class similarity, while at the same time maximizes the between-class similarity. For the embeddings whose corresponding T-F bins belong to the same speaker, the similarity between them should be large and vice versa. This method therefore creates a high-dimensional representation of the mixture audio that results in better segmentation and separation of speakers.

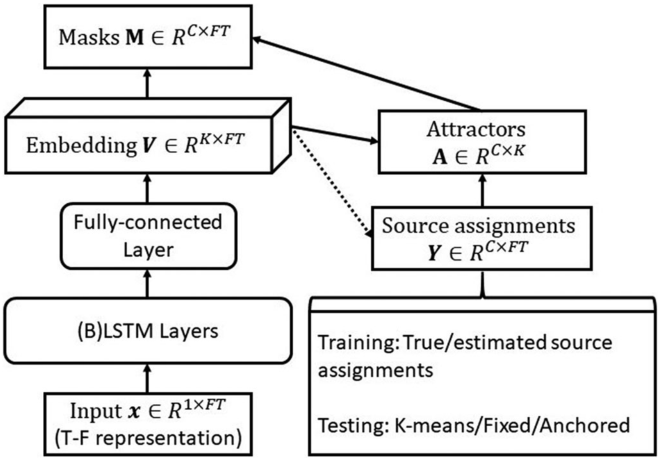
Masks $\mathbf{M} \in R^{C\times FT}$
Embedding $\boldsymbol{V} \in R^{K\times FT}$
Attractors $\mathbf{A} \in R^{C\times K}$
Fully-connected Layer
Source assignments $\boldsymbol{Y} \in R^{C\times FT}$
(B)LSTM Layers
Training: True/estimated source assignments
Testing: K-means/Fixed/Anchored
Input $\boldsymbol{x} \in R^{1\times FT}$ (T-F representation)

Fig. 1. The architecture of DANet. The mixture audio signal is projected into a high-dimensional embedding space. The embeddings of time-frequency bins of each speaker are pulled toward reference points called the attractors. During the training phase, the attractors are formed using true or estimated (in anchored DANet) speaker assignments. During the test phase, the attractors are formed in three alternative ways using: unsupervised clustering, fixed points, and anchor points.

III. Deep Attractor Network

In this section, we introduce the deep attractor network (DANet) [24] and compare it to DPCL [27] and PIT [34]. We then discuss three methods for estimating the parameters of the network during test phase and discuss their pros and cons. We further address the limitation of DANet [24] by presenting a new framework called ADANet. Fig. 1 shows the flowchart of the overall system. Note that in this section we use the same notations as in Section II.

A. Model Definition

Following the main concept of embedding learning with neural network, DANet generates a K-dimensional embedding vector for each of the T-F bins in the mixture magnitude spectrogram

$$\mathbf{V} = f(\mathbf{x}) \tag{6}$$

where $\mathbf{V} \in \mathbb{R}^{K\times FT}$ is the embedding matrix containing all the K-dimensional embeddings for the T-F bins and f is the mapping function implemented by the neural network. Instead of using affinity matrices to measure the similarity between embeddings [27], DANet uses reference points (*attractors*) in the embedding space to calculate similarity as well as for mask estimation.

An *attractor* $\mathbf{a}_i \in \mathbb{R}^{1\times K}$ is a vector in the embedding space that represents a specific speaker, such that all the T-F bins belonging to that speaker are pulled toward the corresponding attractor. The attractors are formed during the training phase by calculating the weighted average of embeddings that belong to each speaker:

$$\mathbf{a}_i = \frac{\mathbf{y}_i \mathbf{V}^\top}{\sum_{f,t} \mathbf{y}_i}, \quad i = 1, 2, \ldots, C \tag{7}$$

where $\mathbf{y}_i \in \mathbb{R}^{1 \times FT}$ denotes the speaker assignment for each T-F bin, which can be either the IBM or IRM in this case (Eqn. 5). Since the attractors represent the centroid of each speaker in the embedding space, averaging over the embeddings that correspond to the most salient T-F bins (i.e. the T-F bins with highest power) may lead to a more robust estimation of the attractors. We therefore apply a threshold to the mixture power spectrogram and create a binary threshold vector $\mathbf{w} \in R^{1 \times FT}$ that filters out the T-F bins with low power. Given a parameter ρ, the threshold vector $\mathbf{w}$ is defined as

$$\mathbf{w}_{ft} = \begin{cases} 1, & \text{if} \quad \mathbf{x}_{ft} > \rho \\ 0, & \text{else} \end{cases} \tag{8}$$

The attractors are then estimated as follows:

$$\mathbf{a}_i = \frac{(\mathbf{y}_i \odot \mathbf{w}) \mathbf{V}^\top}{\sum_{f,t} (\mathbf{y}_i \odot \mathbf{w})}, \quad i = 1, 2, \ldots, C \tag{9}$$

After the generation of the attractors, DANet calculates the similarity between the embedding of each T-F bin and each attractor:

$$\mathbf{d}_i = \mathbf{a}_i \mathbf{V}, \quad i = 1, 2, \ldots, C \tag{10}$$

where $\mathbf{d}_i \in \mathbb{R}^{1 \times FT}$ denotes the distance of each T-F bin in the embedding space from the attractor i. This distance $\mathbf{d}_i$ is small for the embeddings that are close to an attractor, and is large for points that are far away. The mask for each speaker $\hat{\mathbf{m}}_i \in \mathbb{R}^{1 \times FT}, i = 1, \ldots, C$ is then estimated by normalizing the similarity distance using a nonlinear function to constrain the mask to the range $[0, 1]$

$$\hat{\mathbf{m}}_i = \mathcal{H}(\mathbf{d}_i), \quad i = 1, 2, \ldots, C \tag{11}$$

where $\mathcal{H}$ is a nonlinear function which can be either the Softmax or Sigmoid function that is applied to each element of $\mathbf{d}_i$:

$$\begin{cases} Softmax(\mathbf{d}_{i,ft}) = \frac{e^{\mathbf{d}_{i,ft}}}{\sum_{i=1}^{C} e^{\mathbf{d}_{i,ft}}} \\ Sigmoid(\mathbf{d}_{i,ft}) = \frac{1}{\sum_{i=1}^{C} (1 + e^{-\mathbf{d}_{i,ft}})} \end{cases} \tag{12}$$

The neural network is then trained by minimizing a standard L^2 reconstruction error as the objective function

$$l = \frac{1}{C} \sum_i \| \mathbf{x} \odot (\mathbf{m}_i - \hat{\mathbf{m}}_i) \|_2^2 \tag{13}$$

where $\mathbf{m}_i$ and $\hat{\mathbf{m}}_i \in \mathbb{R}^{1 \times FT}$ are the ideal and estimated target masks for the speakers. Since the masks are a function of both attractors and embeddings, optimizing the reconstruction error (Eqn. 13) forces the network to pull the embeddings that belong to the same speaker together and place the attractors far from each other. The initial embeddings and attractors are randomly distributed in the embedding space, and as the training continues, the attractors gradually become separated and pull the embeddings to a proper space that can separate different speakers.

The permutation of the speakers in the outputs is determined by the permutation of the attractors in the embedding space. This is further determined by the permutation of the given speaker assignment vectors $\mathbf{y}_i$. Hence, once the permutation of the target masks $\mathbf{m}_i$ matches the permutation of speaker assignment function $\mathbf{y}_i$, we no longer have the permutation problem in DANet. On the other hand, since the number of the speakers is determined by the number of attractors which is a function of $\mathbf{y}_i$, the output dimension problem is solved in DANet framework, since the number of attractors can change dynamically without changing the network structure.

B. Relation to DPCL and PIT

Since DANet shares the same network structure as DPCL [27], it is important to illustrate the difference between DANet and DPCL. In contrast to DPCL, DANet directly optimizes the reconstruction error with a computationally simpler objective function rather than the calculation of affinity matrices in DPCL. Moreover, the direct mask estimation allows it to use flexible similarity measurements and target masks, including phase-aware [48] and phase-sensitive mask [40].

On the other hand, when the attractors are considered as the trained weights of the network instead of dynamically formed by the embeddings (Eqn. 7 & 9), DANet reduces to a classification network [18], [19] and equation (10) is equivalent to a linear fully-connected layer. In this case, permutation invariant training becomes necessary since the masks are no longer linked to a speaker. However, the dynamic formation of the attractors in DANet allows utterance-level flexibility in the generation of the embeddings. Moreover, DANet does not assume a fixed number of outputs, since the number of attractors is decided by the size of the speaker assignment function Y during training.

IV. Estimation of the Attractor Points

As described in equations (7) & (9), the actual speaker assignment (e.g. using the IBM or IRM methods, Eqn. 5) is necessary to form the attractors during the training phase. However, this information is not available during the test phase, causing a mismatch between training and test phases. In this section, we propose several methods for estimating the location of the attractors during the test phase. The first two methods were introduced in [24]. Here we discuss their limitation and propose a new method for estimating the attractor points called Anchored DANet (ADANet), an extension that enables direct attractor estimation and mask generation for both training and test phases.

A. Forming the Attractors Using Clustering

The simplest method to form the attractors in the embedding space is to use an unsupervised clustering algorithm such as K-means on the embeddings to determine the speaker assignment (DANet-Kmeans in Fig. 4 and Section V). This method is similar to [27]. In this case, the centers of the clusters are treated as the attractors for mask generation. Fig. 2 shows an example of this method, where the crosses represent the centers of the two clusters, which are also the estimated attractors.

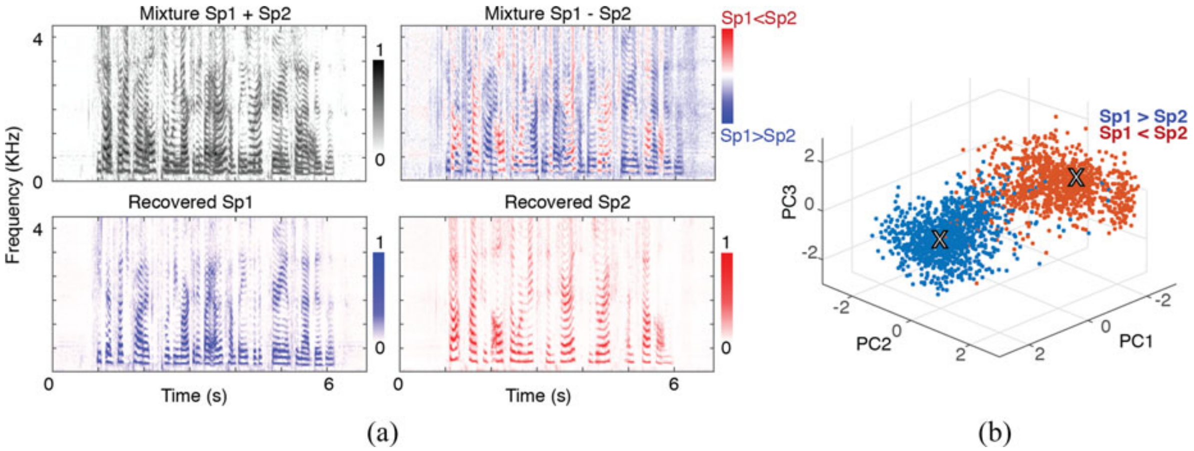

Fig. 2. a) An example of the mixture spectrogram and the recovered speakers. b) Location of T-F bins in embedding space. Each dot visualizes the first three principle components of the embedding of one T-F bin, where colors distinguish the relative power of speakers in that bin. Attractor points are marked with an X.

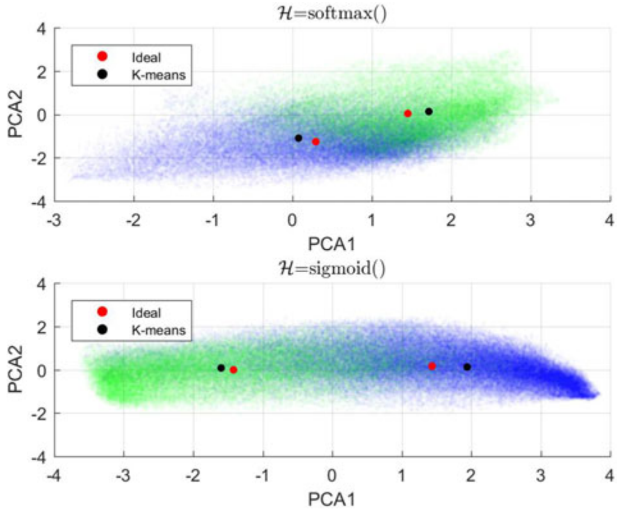

Fig. 3. The gap between the true location of attractors (red dots) and the estimated location using K-means clustering (black dots) in the embedding space, for two networks with softmax and sigmoid nonlinearities. Blue and green dots show the PCA projection of the embedding of T-F bins corresponding to speakers one and two.

B. Fixed Attractor Points

While there is no direct constraint on the location of the attractors in the embedding space, we have found empirically that the location of the attractor points are relatively constant across different mixtures. Fig. 5 shows the location of attractors for 10000 different mixtures, with each opposite pair of dots corresponding to the attractors for the two speakers in a given mixture. Two pairs of attractors (marked as A1 and A2) are automatically discovered by the network. Based on this observation, we propose to first estimate all the attractors in the training phase for different mixtures, and subsequently use the mean of those attractors as the fixed attractors during the test phase (DANet-Fixed in Fig. 4 and Section V). The advantage of using fixed attractors is that it removes the need for the clustering step, allowing the system to directly estimate the mask for each time frame and enabling real-time implementation.

C. Anchored DANet (ADANet)

While both clustering and fixed attractor method can be used during the test phase, these approaches have several limitations. For clustering-based estimation, the K-means step increases the computational cost of the system and therefore increases the run-time delay. Additionally, the centers of the clusters are not guaranteed to match the true locations of the attractors as the (weighted) averages of the corresponding embeddings. This potential difference between the true and estimated attractors causes a mismatch in the mask formation between the training and test phases. One such instance is shown in Fig. 3, where the embedding space is visualized using its first two principle components. The locations of true and estimated attractors are plotted in yellow and black, and the distance between the two shows the mismatch between the true and estimated attractor locations. As suggested by equation (11), this mismatch changes the mask estimated for each source and thus reduces the accuracy of the separation. We refer to this problem as the *center mismatch problem*, which is caused by the unknown speaker assignment during the test phase. Using fixed attractors on the other hand relies on the assumption that the location of the attractors in training and test phases are similar. This, however, may not be the case if the test condition is significantly different from the training condition. Another drawback of fixing the number and location of attractors is the lack of flexibility when dealing with mixtures with variable number of speakers.

To remedy this problem, we propose the Anchored DANet (ADANet), in which several trainable reference points in the embedding space (anchors) are used to estimate the speaker assignment $\mathbf{Y}$ in both training and test phases (Fig. 4). The estimated assignments are then used to find the attractors. This removes the need for true speaker assignment also during the training, and thus removes the *center mismatch problem*.

Similar to the original DANet [24], a K-dimensional embeddings $\mathbf{V}$ for the T-F bins are first generated (Eqn. 6). ADANet first creates N randomly initialized, trainable points in the embedding space $\mathbf{V}$, which are denoted by $\mathbf{b}_j \in \mathbb{R}^{1\times K}$, $j = 1, 2, \ldots, N$, which we call *anchor points*. The number of

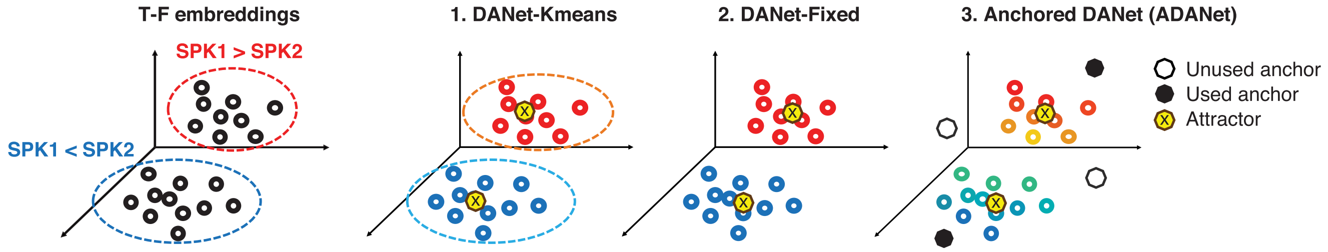

Fig. 4. Different methods for estimating the attractors in DANet. During the test phase, the attractors can be found in three ways: 1) by clustering the embeddings, 2) using fixed attractors calculated from the training set, and 3) first using anchor points to calculate the speaker assignments, followed by the attractor estimation given the speaker assignments.

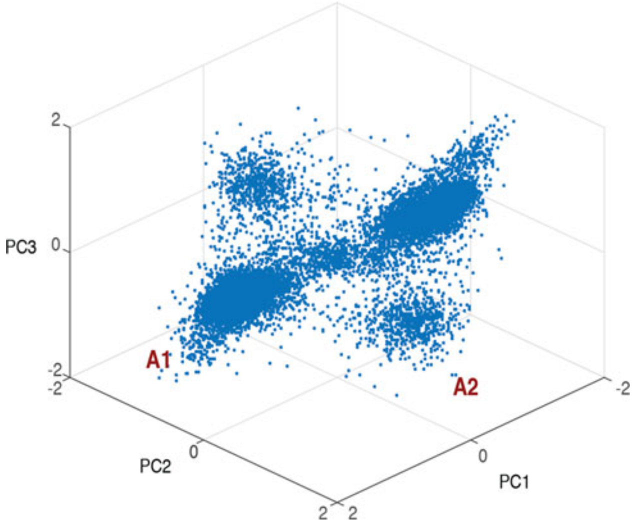

Fig. 5. Location of attractor points in the embedding space in a sigmoid network. Each dot corresponds to one of the 10000 mixtures sounds, visualized using the first three principal components. Two distinct attractor pairs are visible (denoted by A1 and A2).

anchors N is chosen to be no smaller than the number of speakers in the mixture signals in the training set, C. This ensures that the network will have enough capacity to deal with different number of speakers in one general network. In a mixture which contains C speakers, we first choose all the C combinations of the N anchor points $[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N]$, resulting in total of $\binom{N}{C}$ combinations denoted by $\mathbf{L}_p \in \mathbb{R}^{C\times K}$, $p = 1, 2, \ldots, \binom{N}{C}$, where $\binom{N}{C}$ is the standard binomial coefficient. We then find the distance of the embeddings from the anchor points in each subset $\mathbf{L}_p$, and use it to estimate the speaker assignment:

$$\mathbf{D}_p = \mathbf{L}_p\mathbf{V}, \quad p = 1, 2, \ldots, \binom{N}{C} \tag{14}$$

$$\hat{\mathbf{Y}}_p = Softmax(\mathbf{D}_p) \tag{15}$$

where $\mathbf{D}_p \in \mathbb{R}^{C\times FT}$ is the distance between embeddings and each of the C anchors in subset p, and $\hat{\mathbf{Y}}_p \in \mathbb{R}^{C\times FT}$ is the estimated speaker assignment for the corresponding subset. The Softmax function (Eqn. 12) is used to increase the dynamic range of the weights so that they are pushed closer to 0 or 1. The C attractors for each anchor subset are then calculated using the estimated speaker assignment according to equation (7) or (9), leading to $\binom{N}{C}$ sets of attractors $\mathbf{A}_p \in \mathbb{R}^{C\times K}$. The set with the minimum in-set similarity (i.e. largest in-set distance between attractors) is selected for mask estimation

$$\mathbf{S}_p = \mathbf{A}_p\mathbf{A}_p^{\top}, \quad \mathbf{S_p} \in \mathbb{R}^{C\times C} \tag{16}$$

$$s_p = max\{(\mathbf{s}_{p_{ij}})\}, \quad i \neq j \tag{17}$$

$$\hat{\mathbf{A}} = \underset{\mathbf{A}_p}{\arg\min}\{s_p\}, \quad p = 1, 2, \ldots, \binom{N}{C} \tag{18}$$

where s_p is a scalar that represents the maximum similarity between any two attractors in $\mathbf{A}_p$, and $\mathbf{A} \in \mathbb{R}^{C\times K}$ is the set of attractors with smallest in-set similarity among all $\binom{N}{C}$ attractor sets. Given the attractors $\mathbf{A}$, the mask estimation is done in the same manner as equation (10) & (11). Fig. 6 shows examples of the position of the embeddings and the anchor points in a six-anchor point network for four different mixture conditions (two two-speaker mixtures and two three-speaker mixtures), where different sets of anchor points are chosen for estimation of attractors in different conditions. Note that the same network is used for all the mixtures, and training of the network is described in Section V.

The entire process of ADANet can be represented as a generalized Expectation-Maximization (EM) framework, where the speaker assignment calculation is the "Expectation step" and the following attractor formation step is the "Maximization step." Hence, the separation procedure of ADANet can be viewed as a single EM iteration. Note that this method is similar to the unfolded soft-clustering subnetwork proposed in [33] but the parameters (i.e., statistics) for the clusters here are dynamically defined by the embeddings. Thus, they do not increase the total number of parameters in the network. Moreover, this approach also allows utterance-level flexibility in the estimation of the clusters.

D. Attractor Formation Summary

Compared to the clustering and fixed attractor methods, ADANet eliminates the need for true speaker assignment during both training and test phases. The *center mismatch problem* no longer exists because the attractors are determined by the anchor points, which are trained in the training phase and fixed in the test phase. The mask generation process is therefore matched during both training and test phases. Fig. 4 shows the difference between various ways of attractor calculation. The fixed attractor method enables real-time processing and generates the masks without the extra clustering step, but is sensitive to the mismatch between training and test conditions. ADANet framework solves the *center mismatch problem* and allows direct mask

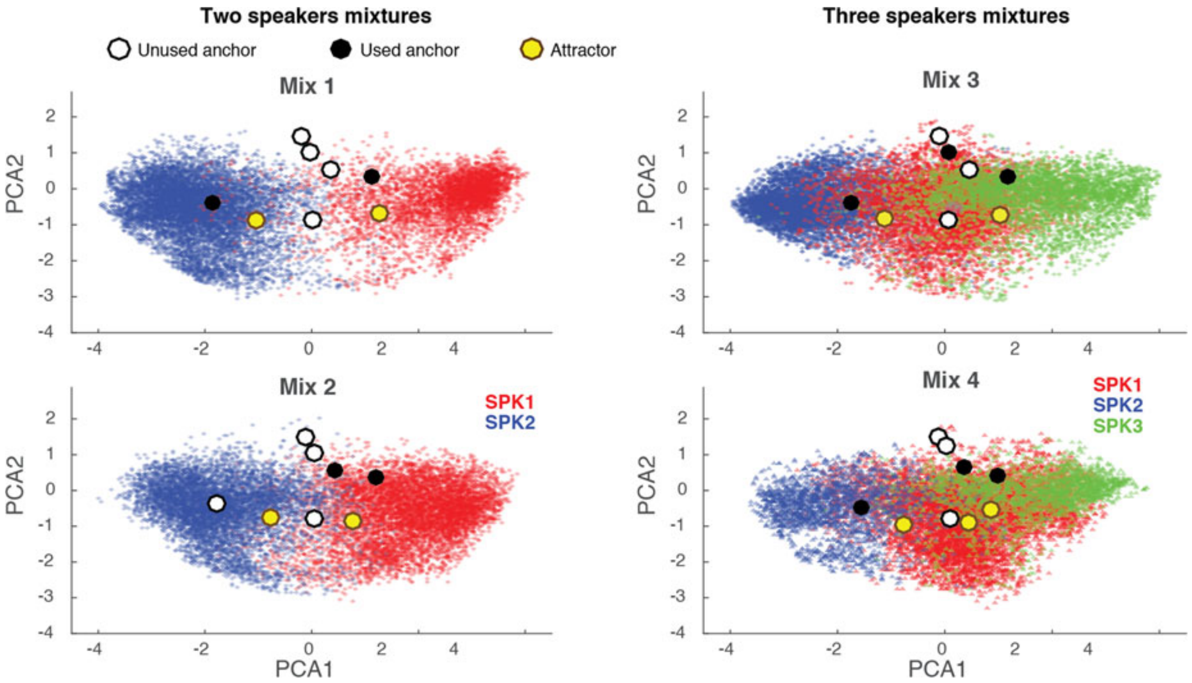

Fig. 6. The PCA projection of the embeddings of the speakers (SPK1..3), anchor points, and attractor points (yellow) for two different two-speaker mixtures, and two different three-speaker mixtures in a single six-anchor point DANet. As can be seen, different anchor points are selected for different mixtures, illustrating the flexibility and stability of ADANet when dealing with different mixture conditions.

generation in both training and test phase, however it increases the computational complexity since all $\binom{N}{C}$ subsets of anchors require one EM iteration. Moreover, since the correct permutation of the anchors is unknown, permutation invariant training in [28] is required for training the ADANet.

V. Experiments and Analysis

We evaluate our proposed model on the task of single-channel two and three speaker separation. Example sounds can be found here [49].

A. Data

We use the WSJ0-2mix and WSJ0-3mix datasets introduced in [27] which contains two 30 h training sets and a 10 h validation sets for the two tasks. These tasks are generated by randomly selecting utterances from different speakers in the Wall Street Journal (WSJ0) training set si_tr_s and mixing them at various signal-to-noise ratios (SNR) randomly chosen between 0 dB and 5 dB. Two 5 h evaluation sets are generated in the same way, using utterances from 16 unseen speakers from si_dt_05 and si_et_05 in the WSJ0 dataset. All data are resampled to 8 kHz to reduce computational and memory costs. The log magnitude spectrogram serves as the input feature, computed using short-time Fourier transform (STFT) with 32 ms window length (256 samples), 8 ms hop size (64 samples), and the square root of Hanning window.

B. Evaluation Metrics

We evaluated the separation performance on the test sets using three metrics: signal-to-distortion ratio (SDR) [50] for comparing with PIT models in [28], [34], scale-invariant signal-to-noise ratio (SI-SNR) to compare with DPCL models in [33], and PESQ score [51] for evaluation of the speech quality. The SI-SNR, proposed in [33], is defined as:

$$\mathbf{s}_{target} := \frac{\langle \hat{\mathbf{s}}, \mathbf{s} \rangle \mathbf{s}}{\|\mathbf{s}\|^2} \tag{19}$$

$$\mathbf{e}_{noise} := \hat{\mathbf{s}} - \mathbf{s}_{target} \tag{20}$$

$$\text{SI-SNR} := 10\, log_{10} \frac{\|\mathbf{s}_{target}\|^2}{\|\mathbf{e}_{noise}\|^2} \tag{21}$$

where $\hat{\mathbf{s}} \in \mathbb{R}^{1\times t}$ and $\mathbf{s} \in \mathbb{R}^{1\times t}$ are the estimated and original clean source respectively, t denotes the length of the signals, and $\|\mathbf{s}\|^2 = \langle \mathbf{s}, \mathbf{s} \rangle$ denotes the power of the signal. $\hat{\mathbf{s}}$ and $\mathbf{s}$ are both normalized to have zero-mean to ensure scale-invariance.

C. Network Architecture

The network contains 4 Bi-directional LSTM [52] layers with 600 hidden units in each layer. The embedding dimension is set to 20 according to [27], resulting in a fully-connected feed-forward layer of 2580 hidden units (20 × 129 as K × F) after the BLSTM layers. Adam algorithm [53] is used for training, with the learning rate starting at $1e^{-3}$ and then halved if no best validation model is found in 3 epochs. The total number of epochs is set to 100, and we used the cost function in equation (13) on the validation set for early stopping. The criterion for early stopping is no decrease in the loss function on validation set for 10 epochs. WFM (Eqn. 5) is used as the training target.

We split the input features into non-overlapping chunks of 100-frame and 400-frame length as the input to the network with a curriculum training strategy [54]. We first train the network with 100-frame length input until converged, and then

TABLE I
SI-SNR IMPROVEMENT (SI-SNRI) IN DECIBEL ON WSJ0-2MIX WITH DIFFERENT ATTRACTOR FORMATION STRATEGIES

Method	$\mathcal{H}$	K-means	ideal
DANet	Sigmoid	9.3	9.6
DANet-90%	Sigmoid	9.5	9.7
DANet-90%-Fixed	Sigmoid	9.2	-
DANet-90%*	Sigmoid	**10.3**	10.6
DANet-90%	Softmax	9.4	9.9
DANet-90%*	Softmax	10.0	10.8

TABLE II
SI-SNR IMPROVEMENTS ON THE WSJ0-2MIX IN ADANET WITH VARYING NUMBER OF ANCHOR POINTS

# of anchors	SI-SNRi
2	9.3
4	9.5
6	9.6
6*	10.1
6-do*	**10.4**

continue training the network with 400-frame long segments. The initial learning rate for 400-frame length training is set to be $1e^{-4}$ with the same learning rate adjustment and early stopping strategies. For comparison, we trained the networks with and without curriculum training. We also studied the effect of dropout [55] during the training of the network by which was added with probability of 0.5 to the input to each of the BLSTM layers. During test phase, the number of speakers is provided to both DANet and ADANet in all the experiments except for the mixed number of speaker experiment shown in Table V.

D. Results

We first examined the effect of choosing different threshold values for attractor estimation. Table I shows the results for different thresholds values (Eqn. 8) in attractor formation, as well as the effect of the nonlinearity functions used for mask generation (Eqn. 11). The 90% suffix denotes a threshold ρ that keeps the top 90% of T-F bins in the mixture spectrogram in equation (8). Notation '-Kmeans' refers to clustering based attractor formation (Section IV-A), and '-Fixed' indicates networks with fixed attractors (Section IV-B). The suffix '-do' indicates training with dropout and the superscript '*' denotes curriculum training.

The results in table I show that using the %90 threshold leads to better performance compared to no threshold, which indicates the importance of accurate estimation of attractors for mask generation. Moreover, we observe that although Softmax with ideal speaker assignment leads to a higher performance than Sigmoid, the performance with K-means for Softmax networks are worse than Sigmoid networks. Our results also show that the K-means performance in the Softmax network is highly dependent on how the network is optimized (i.e. different network trained with different initial values may have very different performance) with a similar level of validation error. This observation is supported by our assumption that Softmax is less sensitive to the distance when the embedding dimension N is high. As a result, adding a training regularization such as dropout does not guarantee improved performance for K-means clustering, which is also a disadvantage of a network architecture that does not directly optimize the mask generation.

Table II shows the effect of changing the number of anchors in ADANet with Softmax for mask generation. All the networks hereafter apply a 90% threshold for estimation of attractors. As the number of anchors increases, the performance of the network consistently improves. It confirms that the increased flexibility in the choice of the anchors help the final separation performance. Unlike Softmax DANet with K-means, adding dropout in BLSTM layers here always increases the performance. This shows the advantage of the framework without using the post-clustering step.

TABLE III
COMPARISON OF DANET AND ADANET WITH OTHER METHODS ON WSJ0-2MIX

Method	Stages	SI-SNRi	SDRi	PESQ
DPCL [27]	1	-	5.8	-
uPIT-BLSTM [34]	1	-	9.4	-
DPCL++ [33]	2	**10.8**	-	-
uPIT-BLSTM-ST [34]	2	-	10.0	-
DANet-Kmeans*	1	10.0	10.3	2.64
DANet-Fixed*	1	9.9	10.2	2.57
ADANet-6-do*	1	10.4	**10.8**	**2.82**
Mixture	-	-	-	2.01
WFM	-	13.9	14.2	3.66

TABLE IV
COMPARISON OF DANET AND ADANET WITH OTHER METHODS ON WSJ0-3MIX

Method	Stages	SDRi	SI-SNRi	PESQ
uPIT-BLSTM [34]	1	7.4	-	-
uPIT-BLSTM-ST [34]	2	7.7	-	-
DPCL++ [33]	2	-	7.1	-
DANet-Kmeans*	1	8.9	8.6	1.92
ADANet-6-do*	1	**9.4**	**9.1**	**2.16**
Mixture	-	-	-	1.65
WFM	-	15.0	15.3	3.40

Tables III & IV compare our method with other speaker-independent techniques in separation of two and three speaker mixtures. DPCL [27] is the original deep clustering model with K-means clustering for mask generation, and DPCL++ [33] is an extension of deep clustering that directly generates the masks with soft-clustering layers, and further improve the masks with a second stage enhancement network. uPIT-BLSTM [34] is the utterance level PIT model that applies PIT on the entire utterance using deep BLSTM network, and uPIT-BLSTM-ST contains a tandem second-stage mask enhancement network for performance improvement. The DPCL and DPCL++ methods also applied a similar curriculum training strategy, while the uPIT-BLSTM and uPIT-BLSTM-ST methods only used standard training. In both tables, we categorize the methods into single-stage and two-stage systems for better comparison, where the two-stage systems have a second-stage enhancement

TABLE V
SDR IMPROVEMENTS (dB) FOR 2 AND 3 SPEAKER MIXTURES TRAINED ON BOTH WSJ0-2MIX AND WSJ0-3MIX

Method	Stages	2 Spk	3 Spk
uPIT-BLSTM [34]	1	9.3	7.1
uPIT-BLSTM-ST [34]	2	10.1	7.8
DANet-Kmeans*	1	10.2	5.3
ADANet-6-do*	1	**10.4**	**8.5**

network. The ADANet system in this comparison utilizes six anchors.

In WSJ0-2mix test set (Table III), the K-means DANet outperforms all the previous one-stage systems and the two-stage uPIT-BLSTM-ST system. The ADANet with six anchors has the best performance among all one-stage systems, and is only slightly worse than the two-stage DPCL++ system. In WSJ0-3mix test set (Table IV), both K-means DANet and six-anchor ADANet outperform all previous systems with either one-stage or two-stage configuration, and ADANet also performs significantly better than K-means DANet. The PESQ scores for DANet and ADANet in WSJ0-2mix show significant improvement upon mixture, while in WSJ0-3mix the gaps between WFM and the models are larger. This is expected since the speaker with lowest energy is harder to separate than the two speaker cases, which may lead to a lower PESQ score.

Table V shows the results in the speaker separation task where the same network is used to separate both two and three speaker mixtures. The networks are first trained on three-speaker mixtures until convergence, and then continued training using both two- and three-speaker mixtures. For K-means DANet, the information about the number of speakers is given during both training and test phases, and for ADANet, we use a training strategy similar to [34], where the network always uses three anchors to generate three masks, and an auxiliary output mask of all zero entries is concatenated to the target masks in the two-speaker cases. During test phase, no information about the number of speakers is provided, and a simple energy-based detector is used to detect the number of speakers. If the power in an output is 20 dB less than the other outputs, it is subsequently discarded.

Table V shows that K-means DANet performs well on the two-speaker separation task, but has worse performance on three-speaker separation. This may be due to the less separated embeddings in two-speaker cases for Softmax function discussed in Table I. However, with the network configuration in ADANet, the performance is significantly better than both one-stage and two-stage PIT systems. Moreover, ADANet successfully detects the correct number of outputs in all the 3000 utterances in the WSJ0-2mix test set, showing that appending a zero-mask to the target masks of two speaker mixtures during training enables the network to learn a silent mask under two-speaker cases without sacrificing the performance in three-speaker separation. This also indicates that ADANet can automatically select proper anchors for speaker assignment estimation in mixtures with different number of sources (Fig. 6).

VI. CONCLUSION

In this paper, we introduce the deep attractor network (DANet) for single-microphone speech separation. We discussed its advantages and drawbacks, and proposed an extension, Anchored DANet (ADANet) for better and more stable performance. DANet extends the deep clustering framework by creating attractor points in the embedding space which pull together the embeddings that belong to a specific speaker. Each attractor in this space is used to form a time-frequency mask for each speaker in the mixture. Directly minimizing the reconstruction error allows better optimization of the embeddings. We explored the mismatch problem of DANet during training and test phase, which is resolved in ADANet to allow direct mask generation in both training and test phases. This removes the post-clustering step for mask estimation. The ADANet approach provides a flexible solution and a generalized Expectation-Maximization strategy to deterministically locate the attractors from the estimated speaker assignment. This approach maintains utterance-level flexibility in attractor formation, and can also generalize to changing signal conditions. Experimental results showed that compared with previous state-of-art systems, both DANet and ADANet have comparable or better performance in both two- and three-speaker separation tasks.

The future works include exploring ways to incorporate speaker information in the estimation of anchor points and in the generation of the embeddings. These aspects can result in both speaker-dependent and speaker-independent separation by the same network. Separating more sources beyond speech signals by creating a larger and more informative embedding space is also an interesting and important topic, which prompts the possibility of designing a universal acoustic source separation framework.

ACKNOWLEDGMENT

The authors would like to thank Drs. John Hershey and Jonathan Le Roux of Mitsubishi Electric Research Lab, and Dr. Dong Yu of Tencent AI Lab for constructive discussions. Yi Luo and Zhuo Chen contributed equally to this work. This work was funded by a grant from National Institute of Health, NIDCD, DC014279, National Science Foundation CAREER Award, and the Pew Charitable Trusts.

REFERENCES

[1] X.-L. Zhang and D. Wang, “A deep ensemble learning method for monaural speech separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 5, pp. 967–977, May 2016.

[2] J. Du, Y. Tu, L.-R. Dai, and C.-H. Lee, “A regression approach to single-channel speech separation via high-resolution deep neural networks,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 8, pp. 1424–1437, Aug. 2016.

[3] D. S. Brungart, “Informational and energetic masking effects in the perception of two simultaneous talkers,” *J. Acoust. Soc. Am.*, vol. 109, no. 3, pp. 1101–1109, 2001.

[4] M. L. Hawley, R. Y. Litovsky, and J. F. Culling, “The benefit of binaural hearing in a cocktail party: Effect of location and type of interferer,” *J. Acoust. Soc. Am.*, vol. 115, no. 2, pp. 833–843, 2004.

[5] N. Mesgarani and E. F. Chang, “Selective cortical representation of attended speaker in multi-talker speech perception,” *Nature*, vol. 485, no. 7397, pp. 233–236, 2012.

[6] W. Xiong *et al.*, “Achieving human parity in conversational speech recognition,” arXiv:1610.05256, 2016.

[7] Y. Cao, S. Sridharan, and A. Moody, "Multichannel speech separation by eigendecomposition and its application to co-talker interference removal," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 5, no. 3, pp. 209–219, May 1997.

[8] R. M. Toroghi, F. Faubel, and D. Klakow, "Multi-channel speech separation with soft time-frequency masking," in *SAPA-SCALE Conf.*, 2012, pp. 86–91.

[9] S. Arberet, A. Ozerov, R. Gribonval, and F. Bimbot, "Blind spectral-GMM estimation for underdetermined instantaneous audio source separation," in *Proc. Int. Conf. Independent Compon. Anal. Signal Separation*, 2009, pp. 751–758.

[10] S. Choi, A. Cichocki, H.-M. Park, and S.-Y. Lee, "Blind source separation and independent component analysis: A review," *Neural Inf. Process.—Letters Rev.*, vol. 6, no. 1, pp. 1–57, 2005.

[11] G. Hu and D. Wang, "A tandem algorithm for pitch estimation and voiced speech segregation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 2067–2079, Nov. 2010.

[12] K. Hu and D. Wang, "An unsupervised approach to cochannel speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 1, pp. 122–131, Jan. 2013.

[13] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *J. Mach. Learn. Res.*, vol. 7, pp. 1963–2001, 2006.

[14] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. New York, NY, USA: Wiley-IEEE Press, 2006.

[15] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[16] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007.

[17] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 1–12, Jan. 2007.

[18] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2013, pp. 436–440.

[19] Z. Chen, S. Watanabe, H. Erdoğan, and J. R. Hershey, "Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 3274–3278.

[20] T. Hori *et al.*, "The MERL/SRI system for the 3rd chime challenge using beamforming, robust feature extraction, and advanced speech recognition," in *Proc. IEEE Workshop Autom. Speech Recognit. Underst.*, 2015, pp. 475–481.

[21] D. S. Williamson and D. Wang, "Time-frequency masking in the complex domain for speech dereverberation and denoising," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 7, pp. 1492–1501, Jul. 2017.

[22] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 7, pp. 1381–1390, Jul. 2013.

[23] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 12, pp. 2136–2147, Dec. 2015.

[24] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 246–250.

[25] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *Proc. Int. Symp. Music Inf. Retrieval*, 2014, pp. 477–482.

[26] Y. Luo, Z. Chen, J. R. Hershey, J. Le Roux, and N. Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 61–65.

[27] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 31–35.

[28] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 241–245.

[29] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 708–712.

[30] F. Mayer, D. S. Williamson, P. Mowlaee, and D. Wang, "Impact of phase estimation on single-channel speech separation based on time-frequency masking," *J. Acoust. Soc. Am.*, vol. 141, no. 6, pp. 4668–4679, 2017.

[31] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *J. Roy. Stat. Soc. Series C (Appl. Stat.)*, vol. 28, no. 1, pp. 100–108, 1979.

[32] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.

[33] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2016, pp. 545–549.

[34] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "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, Oct. 2017.

[35] P. K. Kuhl, "Human adults and human infants show a perceptual magnet effect for the prototypes of speech categories, monkeys do not," *Attention, Perception, Psychophysics*, vol. 50, no. 2, pp. 93–107, 1991.

[36] P. Iverson and P. K. Kuhl, "Mapping the perceptual magnet effect for speech using signal detection theory and multidimensional scaling," *J. Acoust. Soc. Am.*, vol. 97, no. 1, pp. 553–562, 1995.

[37] P. K. Kuhl, B. T. Conboy, S. Coffey-Corina, D. Padden, M. Rivera-Gaxiola, and T. Nelson, "Phonetic learning as a pathway to language: New data and native language magnet theory expanded (NLM-e)," *Phil. Trans. Roy. Soc. London B: Biol. Sci.*, vol. 363, no. 1493, pp. 979–1000, 2008.

[38] Y. Li and D. Wang, "On the optimality of ideal binary time–frequency masks," *Speech Commun.*, vol. 51, no. 3, pp. 230–239, 2009.

[39] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7092–7096.

[40] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux , "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 708–712.

[41] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003.

[42] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[43] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[44] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[45] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2014, pp. 1724–1734.

[46] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. Nat. Conf. Artif. Intell.*, 2016, pp. 3776–3784.

[47] D. P. W. Ellis, "Prediction-driven computational auditory scene analysis," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 1996.

[48] A. Gaich and M. Pejman, "On speech quality estimation of phase-aware single-channel speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 216–220.

[49] [Online]. Available: naplab.ee.columbia.edu/danet.

[50] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

[51] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)—A new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, vol. 2, pp. 749–752.

[52] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[53] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[54] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[55] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

Authors' photographs and biographies not available at the time of publication.