# Deep Clustering With Constant Q Transform For Multi-Talker Single Channel Speech Separation

Ziqiang Shi, Huibin Lin, Liu Liu, Rujie Liu
Fujitsu Research and Development Center
Beijing, China
shiziqiang@cn.fujitsu.com

Shoji Hayakawa
Fujitsu Laboratories Ltd.
Kawasaki, Japan

Jiqing Han
Harbin Institute of Technology
Harbin, China

***Abstract*—Deep clustering technique is a state-of-the-art deep learning-based method for multi-talker speaker-independent speech separation. It solves the label ambiguity problem by mapping time-frequency (TF) bins of the mixed spectrogram to an embedding space, and assigning contrastive embedding vectors to different TF regions in order to predict the mask of the target spectrogram of each speaker. The original deep clustering transforms the speech into the TF domain through a short-time Fourier transform (STFT). Since the frequency component of STFT is linear, while the frequency distribution of human auditory system is nonlinear. Therefore, we propose to use constant Q transform (CQT) instead of STFT to achieve a better simulation of the frequency resolving power of the human auditory system. The ideal upper bound of signal-to-distortion (SDR) of CQT based deep clustering is higher than that based on STFT. In the same experimental setting on WSJ0-mix2 corpus, we gave a detail description in selecting meta-parameters of CQT for speech separation, and finally the SDR improvements of this method achieved about 1dB better performance than the original deep clustering.**

## I. Introduction

Multi-talker monaural speech separation has a vast range of applications. For example, a home environment or a conference environment in which many people talk, the human auditory system can easily track and follow a target speaker's voice from the multi-talker's mixed voice. In this case, if automatic speech recognition and speaker recognition are to be performed, a clean speech signal of the target speaker needs to be separated from the mixed speech to complete the subsequent recognition work. Thus it is a problem that must be solved in order to achieve satisfactory performance in speech or speaker recognition tasks. There are two difficulties in this problem, the first is that since we don't have any priori information of the user, a truly practical system must be speaker-independent. The second difficulty is that there is no way to use the beamforming algorithm for a single microphone signal. Many traditional methods, such as computational auditory scene analysis (CASA) [1], [2], [3], Non-negative matrix factorization (NMF) [4], [5], and probabilistic models [6], [7], do not solve these two difficulties well.

More recently, a large number of techniques based on deep learning is proposed for this task. These methods can be briefly grouped into three categories. The first category is based on deep clustering (DPCL) [8], [9], which maps the time-frequency (TF) points of the spectrogram into the embedding vectors, then these embedding vectors are clustered into several classes corresponding to different speakers, and finally these clusters are used as masks to inversely transform the spectrogram to the separated clean voices; the second is the permutation invariant training (PIT) [10], [11], which solves the label permutation problem by minimizing the lowest error output among all possible permutations for N mixing sources assignment; the third category is end-to-end speech separation in time-domain [12], [13], which is a natural way to overcome the obstacles of the upper bound source-to-distortion ratio improvement (SDRi) in short-time Fourier transform (STFT) mask estimation based methods and real-time processing requirements in actual use.

This paper is based on the DPCL method [8], [9], which has achieved better results than the traditional method. However, DPCL and its most following work use STFT as front-end. Specifically, the mixed speech signal is first transformed from one-dimensional signal in time domain to two-dimensional spectrum signal in TF domain, and then the mixed spectrum is separated to result in spectrums corresponding to different source speeches by a deep clustering method, and finally the cleaned source speech signal can be restored by an inverse STFT on each spectrum. Since the distribution of the frequency components in STFT are linear, while the human auditory system is nonlinear to frequency perception, thus we hope to replace the STFT front-end with certain coefficients that can imitate human auditory system. There are two popular candidates, which are the Mel-frequency cepstral coefficients (MFCC) and constant Q transform (CQT) [14]. However, the MFCC coefficients are not suitable to be a front-end for DPCL for two reasons, one is that it is difficult to do the inverse transform of MFCC coefficients, the other is that the sampling in the frequency-domain of MFCC is sparse. On the other side CQT with the dense coefficients are an easily reversible nonlinear transform which also very similar to the human auditory system [14]. In this work, we showed that DPCL with CQT as front-end can achieve 1dB better performance in separation than that with STFT as front-end.

The remainder of this paper is organized as follows: Section 2 briefly reviews the DPCL framework. Section 3 describes the definition and implementation of CQT. The detail experimental results and comparisons are presented in Section 4 and the whole work is summarized in Section 5.

## II. Speech separation with deep clustering

The main principle of DPCL is to use a powerful network such as LSTM to learn a high-dimensional embedding for each TF unit such that the embedding vectors belonging to the same speaker are close to each other in the embedding space, and farther otherwise [8], [9]. Then these embedding vectors will be clustered into different classes which corresponding to different speakers. Traditional DPCL uses STFT as front-end, however in fact STFT has linear distribution in frequency components, while CQT ensures a constant Q factor across the entire spectrum and thus gives a higher frequency resolution for low frequencies and a higher temporal resolution for high frequencies. Thus in this work we will use CQT as front-end instead of STFT in DPCL to achieve better performance. We summarize the framework of DPCL with CQT as in Fig. 1. The description of CQT will be reviewed in the next section.

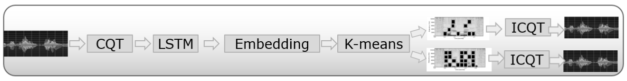

Fig. 1. The framework of deep clustering with CQT

## III. Constant Q transform

### *A. Brief review of CQT*

CQT was proposed by Brown, Judith C. [14] in 1991 to simulate the human auditory system by using a transform with fixed quality factor Q. The quality factor is a concept borrowed from the filter theory, and it is defined as the ratio of the center frequency of the filter to the bandwidth, where the bandwidth refers to the frequency at which is 3 dB less than the highest point of the filter's amplitude-frequency on the characteristic curve. For an ambiguous analogy of the transform domain, the center frequency can be considered as the frequency components of the transform domain, and the bandwidth can be considered as the frequency band of that frequency component. A series of experiments show that CQT achieves better results than STFT in music and speech analysis [14], [16], [17].

The original purpose of introducing CQT is to better analyze the fundamental frequency and the harmonic formant frequency position of the instrument, so as to be able to separate the sound of the musical instrument or achieve a musical instrument effect with better sound characteristics. Therefore, the bandwidth of the CQT frequency component is equivalent to a 1/24th-oct bank filter, as shown in equation (1), where $f_k$ denotes the frequency of the k-th frequency component, B denotes the 1/B octave, and $f_{mtn}$ denotes the minimum frequency of the CQT. The reason why B defaults always to 24 is that studies have shown that the 1/24 octave is similar to the human auditory system, but indeed for the best B is different for different applications.

$$f_k = \left(2^{1/B}\right)^k \cdot f_{mtn} \tag{1}$$

TABLE I. The comparison between CQT and STFT

| | CQT | STFT |
|---|---|---|
| Frequency | $\left(2^{1/B}\right)^k \cdot f_{mtn}$ exponential in k | k$\Delta f$ linear in k |
| Window | Variable=$N[k] = {SR \cdot Q}/{f_k}$ | Constant = N |
| Resolution $\Delta f$ | Variable = $f_k$ / $Q$ | Constant = SR* / N |
| $f_k/\Delta f_k$ | Constant = Q | Variable = k |
| Cycles in Window | Constant = Q | Variable = k |

The time window length corresponding to each frequency component of the STTF is same and fixed, so the frequency components are linearly distributed. Intuitively, it is only necessary to implement CQT by the STTF according to the frequency components of the CQT and take corresponding time windows, as in equation (2). The simple comparison table of CQT and STFF is shown in Table I [14], where SR stands for sampling rate.

$$X[k] = \frac{1}{N[k]}\sum_{n=0}^{N[k]-1} W[k,n]x[n]\exp\{-j2\pi Qn/N[k]\} \tag{2}$$

However, the CQT directly implemented by Equation (2) cannot implement inverse transformation, which greatly limits its scope of use. Velasco, Gino Angelo [16] and Holighaus, Nicki [17] extract the invertible CQT based on the nonstationary Gabor transform (NSGT), and combine STFT and inverse STFT to simplify the computation to improve the transform efficiency. Because of the need to implement an inverse transform, the definition of the DC component and the Nyquist CQT frequency component is increased.

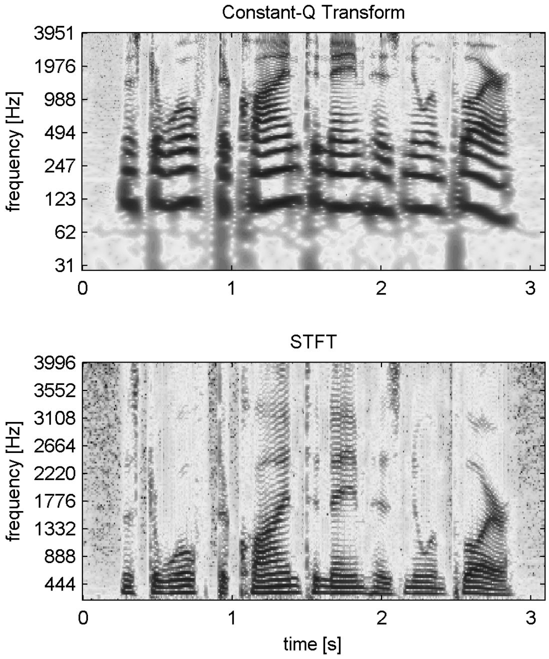

Fig. 2. Spectrograms of ‘22gc0103_1.9955_050c010t_-1.9955.wav’ in wsj0-mix2 dataset. Spectrograms computed with the STFT (top), and with the CQT (bottom).

The relationship between the center frequency and bandwidth of the CQT frequency component are shown in Table II [17], where ξ denotes the frequency and Ω denotes the bandwidth. $\xi_{mtn}$ indicates the lowest frequency of CQT, $\xi_{max}$ indicates the highest frequency of CQT, and $\xi_s$ indicates the sampling rate. Following the tradition, k denotes the index of the CQT frequency component, k = 1, ..., K, K is an integer representing $\xi_{max}$x $\leqslant \xi_k \leqslant \xi_s/2$, where $\xi_s/2$ represents the Nyquist frequency. Using this method can achieve the CQT and ICQT of entire audio or partial audio. In particular, the spectrum in this section refers specifically to the frequency spectrum from the STFT where the frequency component is linear.

TABLE II. THE RELATIONSHIP BETWEEN CQT CENTER FREQUENCY AND BANDWIDTH

| k | $\xi_k$ | $\Omega_k$ |
|---|---|---|
| 0 | 0 | $2\xi_{mtn}$ |
| 1,…..,K | $\xi_{mtn}2^{\frac{k-1}{B}}$ | $\xi_k/Q$ |
| K+1 | $\xi_s/2$ | $\xi_s-2\xi_K$ |
| K+2,…..,2K+1 | $\xi_s-\xi_{2K+2-k}$ | $\xi_{2K+2-k}/Q$ |

According to the basic definition of CQT, the lower the frequency, the larger the bandwidth. However, for the human auditory system, only when the frequency is higher than 500Hz, it is similar to CQT, and the bandwidth below 500Hz is close to smooth. Therefore, in calculating the bandwidth of CQT, a new parameter γ is introduced, and the specific calculation formula is as shown in the following equations.

$$B_k = \alpha f_k + \gamma \quad (3)$$

$$\alpha = 2^{\frac{1}{b}} - 2^{-\frac{1}{b}} \quad (4)$$

where b represents the bandwidth of each octave equivalent filter.

The process of the forward and reverse transformation of CQT will be briefly summarized below. The process of forward transformation is:

1) Obtain the spectrum through STFT.

2) Based on the length of STFT and B, calculate the number of CQT frequency components and the spectral range covered by each CQT frequency component, and the length of the frequency domain window function that simulates the time-domain down-sampling.

3) The spectral data of each CQT frequency component corresponding range is extracted by a window function, and the length of the highest frequency component is achieved by zero padding, to provide information redundancy and matrix output.

4) The CQT data of this frequency component is obtained by using inverse STFT on the zero padded data.

Inverse CQT process:

1) Do STFT to the data of each CQT frequency component and the number of conversion points is the length of the data.

2) According to the time length of each CQT frequency component to simulate the time domain down-sampling, it is taken from the low frequency on the STFT data, and then put into the corresponding spectrum position. Where the length of the spectrum is the length of the data block.

3) After the operation of all CQT frequency components is completed, the frequency spectrum undergoes inverse STFT to obtain the time domain signal recovered by the CQT.

## IV. EXPERIMENTAL RESULTS

### *A. Dataset and neural network*

We evaluated our system on two-speaker speech separation problem using WSJ0-2mix dataset [8], [9], which contains 30 hours of training and 10 hours of validation data. The mixtures are generated by randomly selecting 49 male and 51 female speakers and utterances in Wall Street Journal (WSJ0) training set si_tr_s, and mixing them at various signal-to-noise ratios (SNR) uniformly between 0 dB and 5 dB . 5h of evaluation set is generated in the same way, using utterances from16 unseen speakers from si_dt_05 and si_et_05 in the WSJ0 dataset. To reduce the computational cost, the waveforms were down-sampled to 8 kHz.

We re-implemented the traditional DPCL. The network structure and parameters of DPCL used in this paper is basically consistent with the literature [8], [9]. The neural network for extracting the embedding vector is a 4-layer bidirectional LSTM with 600 cells in the forward and backward directions. Each layer of LSTM introduces dropout of 0.3 but does not introduce recurrent dropout. Finally, an embedding vector is output through a fully connected layer. The network is trained from scratch using the Adam algorithm. The window length is 32ms, the hop size is 8ms, and the square root of the Hamming window is used as the analysis window for traditional DPCL with STFT. 256-point STFT is performed to extract the 129-dimensional log magnitude feature of each frame for BLSTM training.

### *B. The selection of CQT parameters*

In this section, we select the most appropriate CQT configuration parameters by calculating the SDRi upper bound of the separated speech. For the case where the two speaker's speech is mixed into one, the SDRi upper bound of CQT based DPCL is better than the STFT based under the same experimental conditions.

The main parameters used in CQT are B, γ, and the window functions. At the same time, the data block length and the minimum frequency of a single CQT processing will theoretically also have the influence on performance of voice separation. Therefore, we evaluate the influence of various factors on the separation performance by calculating the upper bound of the ideal SDRi. The literature [8], [9] defines the SDR ideal upper bound as follows: compute the ideal mask $a_i^{\mathrm{ibm}} = \delta\left(|s_i| > \max_{j\neq i}|s_j|\right)$ from clean signals that are not mixed compared to the mixed signal, and then the speech signal of each speaker is separated by these ideal mask. The SDR calculated based on these separated signals is called the ideal SDR upper bound. SDRi represents the SDR of the separated voices minus the original SDR without separation. In our computation, the original SDR of the evaluation data set = 0.15dB, which is consistent with the work [8], [9]. The SDRi ideal upper bound of STFT based DPCL is 13.5dB.

In this section five groups of tests are used to analyze the effect of different CQT parameters on the separation performance, which will be introduced one by one. The default parameters are  of B=24, γ=20, window function is Hanning window, the length of CQT processing data block is the length of each segment of mixed audio, and the lowest frequency of CQT is 27.5 Hz.

The first group of tests is to evaluate the effect of the number of equivalent filters B per octave on the separation performance. The other parameters were fixed as γ = 20, the window function was a Hanning window, and the CQT processing data block length was the length of each mixed audio segment. The lowest frequency of CQT is 27.5Hz. Table III shows the results of this set of experiments and shows that B=36 can achieve better separation performance.

TABLE III. THE EFFECT OF THE NUMBER OF EQUIVALENT FILTERS (B) ON THE SEPARATION PERFORMANCE (DB)

| B | 12 | 24 | 36 | 48 | 60 | 72 |
|---|---|---|---|---|---|---|
| SDRi | 13.99 | 14.73 | 14.77 | 14.69 | 14.59 | 14.50 |

The second set of tests evaluated the effect of γ on separation performance. Other parameters were: B=24, window function is Hanning window, CQT processing data block length is the length of each mixed audio, and CQT minimum frequency is 27.5 Hz. Table IV shows the experimental results of this group, where the bandwidth of the equivalent filter of CQT is equal to ERB at γ=26.4. The experimental results show that the larger the γ, the better the separation performance.

TABLE IV. EFFECT OF Γ ON SEPARATION PERFORMANCE (DB)

| γ | 0 | 3 | 10 | 20 | 26.4 | 30 |
|---|---|---|---|---|---|---|
| SDRi | 13.64 | 14.04 | 14.53 | 14.83 | 14.88 | 14.88 |

The third group of tests is to evaluate the influence of the window function on the separation performance. The other parameters are: B=24, γ=20, the length of the CQT processing data block is the length of each mixed audio, and the lowest frequency of the CQT is 27.5 Hz. Table V shows the results of this set of experiments and shows that the window function is a Hamming window for better separation performance.

The fourth group of tests is to evaluate the effect of different data block lengths on the separation performance. The other parameters are fixed as B=24, γ=20, window function is Hanning, and CQT minimum frequency is 27.5 Hz. Table VI shows the experimental results of this group, where seconds indicates the time length of the data block, and all indicates that the data block is not divided. The results show that the longer the data block length, the better the performance.

TABLE V. EFFECT OF WINDOW FUNCTION ON SEPARATION PERFORMANCE

| Window | hanning | cos | rectangle | triangle | hamming |
|---|---|---|---|---|---|
| SDRi | 14.83 | 14.91 | 14.32 | 14.89 | 14.94 |
| Window | Blackman | blackharr | modblackharr | nuttall | nuttall10 |
| SDRi | 14.61 | 14.25 | 14.25 | 14.22 | 14.83 |
| Window | nuttall101 | nuttall20 | nuttall11 | nuttall02 | nuttall30 |
| SDRi | 14.94 | 14.37 | 14.56 | 14.61 | 13.76 |
| window | nuttall21 | nuttall12 | nuttall03 | gauss | wp2inp |
| SDRi | 14.07 | 14.22 | 14.28 | 14.57 | 12.57 |

TABLE VI. EFFECT OF DATA BLOCK LENGTH ON SEPARATION PERFORMANCE

| seconds | 0.125 | 0.25 | 0.5 | 1 | all |
|---|---|---|---|---|---|
| SDRi | 13.58 | 14.23 | 14.52 | 14.69 | 14.83 |

The fifth set of experiments evaluated the effect of increasing the minimum frequency of CQT to 110 Hz on the separation performance. Because the fundamental frequency of the speech is around 200 Hz, considering the bandwidth of the signal, the probability that the effective frequency component of the speech signal lower than 110 Hz is very low. Therefore, by increasing the minimum frequency of CQT, the data amount is reduced by 27.6%, thereby reducing the complexity of deep clustering training. Other parameters are fixed as B=24, γ=20, window function is Hanning, and CQT minimum frequency is 110 Hz. Table VII shows the experimental results of this group. The results are similar to those of the fourth group. The longer the data block length is, the better the performance is. However, the SDRi of all data block lengths is lower than the experimental results of the fourth group, indicating that the minimum frequency of the CQT is raised to 110 Hz or Performance has a negative impact.

TABLE VII. EFFECT OF CQT MINIMUM FREQUENCY TO 110HZ DATA BLOCK LENGTH ON SEPARATION PERFORMANCE

| seconds | 0.125 | 0.25 | 0.5 | 1 | all |
|---|---|---|---|---|---|
| SDRi | 13.50 | 14.08 | 14.35 | 14.50 | 14.58 |

Finally, combining all the experimental results, the best parameters of CQT for speech separation should be B=36, γ=20, window function is Hamming window, CQT minimum frequency is 27.5 Hz, no data block. However, considering the complexity of calculation, training the network requires the unity of data length, and because of the mistakes of previous array experiments, the standard experimental parameters of our CQT are B=36, γ=20, window function is cosine window, CQT minimum frequency It is 110Hz and the data block length is 1 second. The theoretical upper limit for SDRi is based on a CQT framework that is 1 dB higher than the STFT-based framework.

### C. CQT based DPCL vs. conventional DPCL

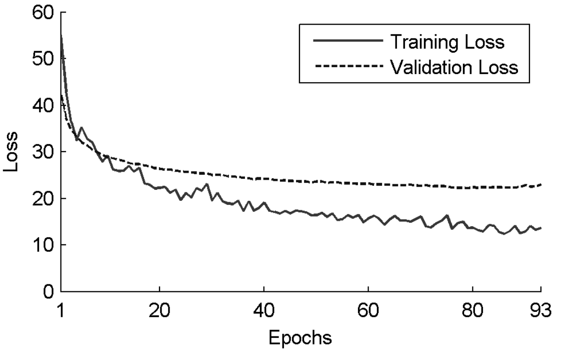

Fig. 3 Loss over epochs on the WSJ0-2mix training and validation sets with CQT based DPCL

Based on the previous section of the experiment using CQT parameters and a deep clustering network, we trained the network and calculated the SDRi of the model. Other network parameters include batch size = 32. Each data block contains 1 second of audio content. The CQT data block size is 126*323, which means that the frequency components of the CQT are 126 in total. Each second contains 323 CQT samples. The training and validation loss is shown in Fig. 3. The performance is shown in the Table VIII. It can be seen that similar to the difference between the ideal SDRi upper bound, CQT based DPCL achieved one 1dB better performance than traditional STFT based DPCL.

TABLE VIII. THE SEPARATION PERFORMANCE (dB) OF CQT BASED DPCL

| DPCL | CQT based | STFT based |
|---|---|---|
| SDRi | 10.7 | 9.6 |

## V. CONCLUSION

In this paper we have proposed a monaural speech separation method based on constant q transform (CQT) and deep clustering. We give a detail description in selection of the meta-parameter of CQT in speech separation. Since CQT ensures a higher frequency resolution for low frequencies and a higher temporal resolution for high frequencies, we achieve better separation results than conventional deep clustering which uses short time Fourier transform (STFT) as front-end. In future we plan to test wether CQT will constantly better than STFT in other separation methods, such as PIT based framework [10], [11].

## ACKNOWLEDGMENT

We would like to thank Jian Wu at Northwestern Polytechnical University and Yi Luo at Columbia University for valuable discussions on WSJ0-mix2 and deep clustering.

## REFERENCES

[1] D. Wang and G. J. Brown, Computational Auditory Scene Analysis: Principles, Algorithms, and Applications. Wiley-IEEE Press, 2006.

[2] Y. Shao and D. Wang, “Model-based sequential organization in cochannel speech,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 289–298, Jan. 2006.

[3] K. Hu and D. Wang, “An Unsupervised Approach to Cochannel Speech Separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 1, pp. 122–131, Jan. 2013.

[4] P. Smaragdis, “Convolutive Speech Bases and Their Application to Supervised Speech Separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 1–12, Jan. 2007.

[5] J. L. Roux, F. Weninger, and J. R. Hershey, “Sparse NMF – half-baked or well done?” Mitsubishi Electric Research Labs (MERL), Tech. Rep. TR2015-023, 2015.

[6] T. Virtanen, “Speech Recognition Using Factorial Hidden Markov Models for Separation in the Feature Space,” in *Proc. INTERSPEECH*, 2006.

[7] M. Stark, M. Wohlmayr, and F. Pernkopf, “Source-Filter-Based Single Channel Speech Separation Using Pitch Information,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 2, pp. 242–255, Feb. 2011.

[8] J. R. Hershey, Z. Chen, and J. Le Roux, “Deep Clustering: Discriminative Embeddings for Segmentation and Separation,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016.

[9] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, “Single-Channel Multi-Speaker Separation using Deep Clustering,” in *Proc. Interspeech*, Sep. 2016.

[10] 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 Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[11] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, “Permutation invariant training of deep models for speaker-independent multitalker speech separation,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017, pp. 241–245.

[12] Yi Luo, and Nima Mesgarani, "TasNet: time-domain audio separation network for real-time, single-channel speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on,* 2018.

[13] Venkataramani, S., J. Casebeer, and P. Smaragdis. “Adaptive Front-ends for End-to-end Source Separation” , in *Workshop for Audio Signal Processing*, NIPS 2017.

[14] J. C. Brown, “Calculation of a constant Q spectral transform,” *J. Acoust. Soc. Am.*, vol. 89, no. 1, pp. 425–434, 1991.

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

[16] G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, “Constructing An Invertible Constant-Q Transform With Nonstationary Gabor Frames,” *Artif. Intell.*, pp. 93–99, 2011.

[17] N. Holighaus, M. Dörfler, G. A. Velasco, and T. Grill, “A framework for invertible, real-time constant-Q transforms,” *IEEE Trans. Audio, Speech Lang. Process.*, vol. 21, no. 4, pp. 775–785, 2013.

[18] Danwei Cai, Zhidong Ni, Wenbo Liu, Weicheng Cai, Gang Li, Ming Li, “End-to-End Deep Learning Framework for Speech Paralinguistics Detection Based on Perception Aware Spectrum.” *INTERSPEECH*, 2017.