AUGMENTED TIME-FREQUENCY MASK ESTIMATION IN CLUSTER-BASED SOURCE SEPARATION ALGORITHMS

Yi Luo *Nima Mesgarani*

Department of Electrical Engineering, Columbia University, New York, NY

ABSTRACT

Time-frequency mask estimation with various clustering approaches has proven effective in solving the audio source separation problem. In this framework, the time-frequency bins of the mixture spectrogram are represented in a high-dimensional embedding space, where various methods can be applied to group the embedded points to calculate either hard or soft source assignments and subsequently the time-frequency masks. However, the mismatch between the assignment algorithm during the training and inference phases in majority of the current approaches leads to a suboptimal solution, because the assignment objective that is used during the training (e.g. ideal binary mask) is not the same as the one used during the inference phase (e.g. k-means clustering). We propose a method to reduce the mismatch between these two conditions where the source embedding is trained such that the source assignment during training and inference phases results in similar outcomes. Our results show that matching the source assignment during training- and inference-phase results in more accurate and consistent mask estimation in the inference phase which significantly improves the source separation accuracy for various hard and soft clustering methods.

Index Terms— Source separation, clustering, mask estimation, deep learning

1. INTRODUCTION

Time-frequency (T-F) masking has long been one of the most prevalent and successful approaches for source separation. Among various methods for estimating the T-F masks, clustering-based or grouping-based approaches have proven their effectiveness and potential under various circumstances [1, 2, 3, 4]. The general framework for clustering-based approaches towards T-F masking is to formulate the mask estimation as a classification problem, where the source assignment labels or probabilities (hard or soft masking) for the T-F bins in the mixture spectrogram represent the corresponding T-F masks.

In recent years, deep learning-based clustering models have greatly advanced the state-of-the-art of this problem [5, 6, 7, 8, 9, 10, 11, 12, 13]. Various neural network architectures and procedures for embedding generation and cluster formulation have been proposed for robust performance and good generalization. However, an important issue for many clustering-based systems is that the training- and inference-phase conditions are mismatched. This may lead to unpredictable failure in inference phase even if the network has been trained properly in the training phase. Moreover, with the sensitivity of most clustering algorithms to their initialization, the performance in inference phase is even harder to guarantee. Several previous studies proposed different objective functions favored by the type of clustering algorithm selected in the inference phase [10], multi-stage learning frameworks [6, 7, 13] or explicit cluster parameter initialization inside the network [9, 12] to alleviate the mismatch, however they were either not straight-forward or greatly increased the system complexity.

We propose a simple but effective method for bridging the performance gap between the training- and inference-phases. The aim for the method is to use the same computation in the training phase as the one it was trained on. The key concept in the method is to train the network such that the ideal binary mask (IBM) is a local minimum of a selected hard clustering algorithm (e.g. K-means) on the generated embeddings. With networks trained with this property, hard T-F masks can be calculated by applying the selected hard clustering algorithm in the inference phase. For soft mask estimation tasks, an extra soft mask estimation objective can be added to the network in a multi-task learning fashion, with the IBM used as the initialization of the soft mask estimation procedures. In the inference phase, the estimated hard mask is used as the initialization for soft mask estimation. This is equivalent to train the embeddings so that they simultaneously follow two probability distributions defined by the hard and soft clustering algorithms respectively, while the hard assignments serve as the cue for initialization in the soft clustering branch. Experiments show that this simple method significantly bridges the performance gap between the training- and inference-phases on various hard and soft clustering algorithms.

The rest of the paper is organized as follows. Section 2 introduces and analyzes the proposed method. Section 3 provides experiment results and discussions. Section 4 concludes the paper.

2. MATCHING TRAINING- AND INFERENCE-PHASE CONDITIONS FOR MASK ESTIMATION

2.1. Problem formulation

In clustering-based T-F masking methods with deep neural networks for embedding generation, the mixture spectrogram $\mathbf{X} \in \mathbb{R}^{T \times F}$ is mapped to a D-dimensional embedding space

$$\mathbf{V} = \mathcal{H}(\mathbf{X}) \tag{1}$$

where $\mathbf{V} \in \mathbb{R}^{D \times TF}$ are the embeddings for the T-F bins, and $\mathcal{H}(\cdot)$ is the mapping function defined by the neural network. The general idea during training phase is that $\mathbf{V}$ should be clustered into C classes, each representing an active source in the mixture. The source assignment labels or probabilities $\mathbf{W}' \in \mathbb{R}^{C \times TF}$ are then treated as the estimated T-F masks. Training objectives are typically designed to favor a selected type of clustering algorithm, such that $\mathbf{W}'$ matches closely to the target T-F masks $\mathbf{W} \in \mathbb{R}^{C \times TF}$. For hard clustering, ideal binary mask (IBM) $\mathbf{B} \in \{0, 1\}^{C \times TF}$ is typically used as the target oracle mask [14], while for soft clustering various other masks can be chosen [15].

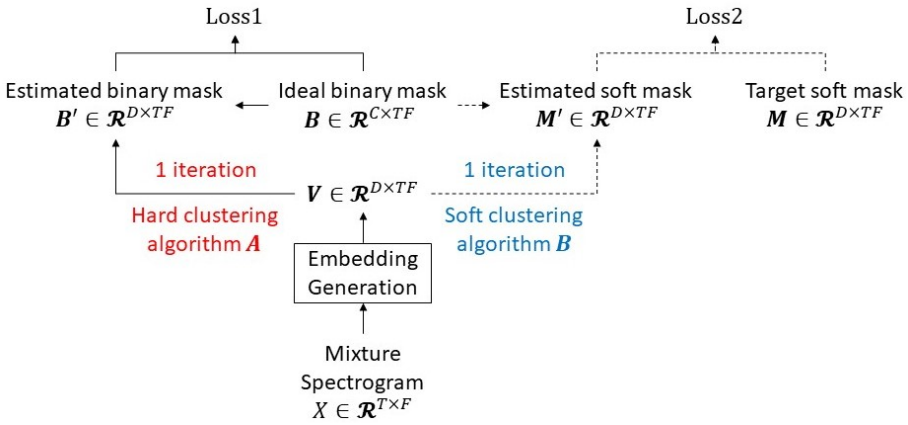

Fig. 1. Flowchart of the proposed training method for clustering-based T-F mask estimation. Embeddings $\mathbf{V}$ are generated from the mixture spectrogram $\mathbf{X}$ through a neural network. For hard mask estimation, one iteration of a selected hard clustering algorithm is performed with ideal binary masks $\mathbf{B}$ as the initialization. For soft mask estimation, the soft masks $\mathbf{M}'$ is estimated with any soft clustering algorithm with $\mathbf{B}$ as the initialization.

2.2. Ideal binary mask as a local minimum

Many widely-used clustering algorithms rely on iterative update rules (e.g. Expectation-Maximization) to find a local minimum. An important property for those iterative rules is their convergence guarantee: the objective function (e.g. log-likelihood in EM) is improved for every iteration, and the algorithm will converge after a finite number of iterations. A typical convergence criteria is that the difference between the source assignments generated from two consecutive iterations is smaller than a pre-defined threshold. In other words, a source assignment $\hat{\mathbf{W}}$ is a local minimum of the clustering problem, if $\hat{\mathbf{W}}'$ is the source assignment after another iteration and satisfies $||\hat{\mathbf{W}}' - \hat{\mathbf{W}}||_2 < \epsilon$ with $\epsilon \in \mathbb{R}^+$ be the threshold. Motivated by this, we can design an objective function such that IBM $\mathbf{B}$ is a local minimum: when initializing a hard clustering algorithm with $\mathbf{B}$ as the source assignments, it should still generate $\mathbf{B}$ as the source assignments after one more iteration. Using $\mathcal{L}_2$-norm as objective, the objective can be written as:

$$\mathcal{L} = ||\mathbf{X} \odot (\mathbf{B} - \mathbf{B}')||_2^2 \tag{2}$$

where $\mathbf{B}'$ is the generated source assignment and $\odot$ represents element-wise multiplication. This is equivalent to a traditional mask approximation objective with IBM as target, but the difference here is that the masks $\mathbf{B}'$ is generated from a standard iteration of the selected hard clustering algorithm. For EM-based hard clustering algorithms such as K-means, it corresponds to the procedure that an M-step is first applied to obtain the parameters of the clustering model, and an E-step is followed to generate the new source assignments (i.e. masks). During inference phase, the selected hard clustering algorithm can be directly applied until convergence.

One issue during training is that binarizing $\mathbf{B}'$ as the output of the M-steps might lead to harder gradient back-propagation, especially with the usage of the non-differentiable *argmax* or *argmin* functions. To alleviate it, we perform a continuous relaxation on $\mathbf{B}'$ to allow proper gradient flow. As examples, we demonstrate how the E-steps of K-means and spherical K-means can be relaxed. Other types of hard clustering algorithms can be relaxed similarly.

2.2.1. *Continuous relaxation for K-means*

The E-step in standard K-means estimates the source assignment of an embedding by choosing the closest cluster centroid:

$$\mathbf{r}_{n,t,f} = argmin\,\{||\mathbf{v}_{t,f} - \boldsymbol{\mu}_{i,n}||_2,\ i = 1, \ldots, C\} \tag{3}$$

where $\mathbf{r}_{n,t,f} \in \mathbb{R}^{C\times 1}$ corresponds to the source assignment of T-F bin {t, f} at n-th iteration, $\mathbf{v}_{t,f} \in \mathbb{R}^{D\times 1}$ is the embedding vector of T-F bin {t, f}, and $\boldsymbol{\mu}_{i,n} \in \mathbb{R}^{D\times 1}$ is the i-th centroid at n-th iteration. To replace the *argmin* function, we relax it with the squared Euclidean distance between $\mathbf{v}_{t,f}$ and all $\boldsymbol{\mu}_{i,n}$:

$$\hat{\mathbf{r}}_{n,t,f} = \{\mathbf{1} - \frac{||\mathbf{v}_{t,f} - \boldsymbol{\mu}_{i,n}||_2^2}{\sum_{i=1}^{C} ||\mathbf{v}_{t,f} - \boldsymbol{\mu}_{i,n}||_2^2},\ i = 1, \ldots, C\} \tag{4}$$

2.2.2. *Continuous relaxation for spherical K-means*

Spherical K-means is designed for clustering embeddings on a unit hypersphere [16]. The standard E-step applies the *argmax* function on the dot product between the embeddings and the centers

$$\mathbf{p}_{n,t,f} = argmax\,\{\boldsymbol{\mu}_{i,n}^T \bar{\mathbf{v}}_{t,f},\ i = 1, \ldots, C\} \tag{5}$$

where $\mathbf{p}_{n,t,f} \in \mathbb{R}^{C\times 1}$ and $\boldsymbol{\mu}_{i,n} \in \mathbb{R}^{D\times 1}$ correspond to the source assignments of T-F bin {t, f} and the i-th cluster centers at n-th iteration respectively, and $\bar{\mathbf{V}} \in \mathbb{R}^{D\times TF}$ is the embedding matrix normalized to unit $\mathcal{L}_2$-norm. We relax the *argmax* function with a Softmax function

$$\mathbf{a}_{n,i,t,f} = \begin{cases} \mathbf{1} - arccos(\boldsymbol{\mu}_{i,n}^T \bar{\mathbf{v}}_{t,f}/\pi), \text{ if } \mathbf{V} \in \mathbb{R} \\ \mathbf{1} - 2 \cdot arccos(\boldsymbol{\mu}_{i,n}^T \bar{\mathbf{v}}_{t,f}/\pi), \text{ if } \mathbf{V} \in \mathbb{R}^+ \end{cases} \tag{6}$$

$$\hat{\mathbf{p}}_{n,i,t,f} = e^{\alpha \mathbf{a}_{n,i,t,f}} \oslash \sum_{i=1}^{C} e^{\alpha \mathbf{a}_{n,i,t,f}} \tag{7}$$

where $\mathbf{A}_{n,i} \in \mathbb{R}^{1\times TF}$ represents the angular similarity between all the embeddings and the i-th center at iteration n, and $\oslash$ is the element-wise division operation. $\alpha \in [1, +\infty)$ is a scalar sharpening the output, which is similar to the gumbel-softmax function [17]. We empirically set $\alpha = 5$ in all our experiments.

2.2.3. Relaxation in multiple iterations

If more than one EM iteration is applied during training phase, only the last E-step should have continuous output to guarantee an identical update rule with the inference phase. In order to properly propagate the gradients in the intermediate steps where a binarized output is required, a simple straight-through estimator [18] can be applied to copy the gradient of the binarized source assignments to their continuous relaxations. Suppose $\mathbf{Z} \in \mathbb{R}^{C \times TF}$ is the relaxed source assignment matrix for an intermediate E-step of any hard clustering algorithm, then the straight-through estimator can be defined as

$$ST(\mathbf{Z}) : gradient\_copy([\mathbf{Z}], \mathbf{Z}) \tag{8}$$

where $[\cdot]$ is the nearest integer function (i.e. rounding function), and *gradient_copy* copies the gradient received by $[\mathbf{Z}]$ directly to $\mathbf{Z}$ for back-propagation. $[\mathbf{Z}]$ is used as the binary assignment matrix to pass to the next M-step. However, stacking multiple iterations has the risk that every two iterations might not be converging. For instance, with two EM iterations in K-means and $\mathbf{R}_{1,2} \in \mathbb{R}^{C \times TF}$ as the corresponding source assignments, optimizing equation 2 with $\mathbf{R}_2$ as the final output does not explicitly constrain convergence between the pairs ($\mathbf{B}$, $\mathbf{R}_1$) and ($\mathbf{R}_1$, $\mathbf{R}_2$). In other words, IBM might not be the local maximum since the steps from $\mathbf{B}$ to $\mathbf{R}_1$ and $\mathbf{R}_1$ to $\mathbf{R}_2$ might not meet the convergence criteria. We will show in section 3.3 that applying more than one iteration does not improve the performance.

2.3. Multi-task learning for soft mask estimation

A multi-task learning framework can be designed for soft T-F mask estimation. Similar to the hard mask estimation, we initialize the soft clustering algorithm with the IBM $\mathbf{B}$ and run 1 iteration for update. During inference time, the binary mask can be estimated from the selected hard clustering algorithm, and then used as the initialization for the soft clustering algorithm for 1 iteration. The multi-task learning objective then becomes:

$$\mathcal{L}_{MT} = ||\mathbf{X} \odot (\mathbf{B} - \mathbf{B}')||_2^2 + ||\mathbf{X} \odot (\mathbf{M} - \mathbf{M}')||_2^2 \tag{9}$$

where $\mathbf{M}$ and $\mathbf{M}'$ are the target and the estimated soft masks respectively. This is equivalent to the assumption that when initialized with IBM, the soft clustering algorithm will generate the target soft mask after one iteration. Figure 1 shows the entire flowchart of the method. Note that the soft clustering algorithm can be any mask estimation module even without the constraint that the summation of the assignments should be 1, which enables the system to estimate any real-valued T-F masks [15] or directly perform magnitude spectrum approximation (MSA) [19].

A natural question arises in the light of equation 2: can soft clustering algorithm be designed in the same way such that a target soft mask $\mathbf{M}$ is the local minimum? Under this assumption, the objective function would simply be

$$\mathcal{L}' = ||\mathbf{X} \odot (\mathbf{M} - \mathbf{M}')||_2^2 \tag{10}$$

We argue that this assumption is too hard or even impossible to achieve. Discretized masks have the advantage that they are more robust to small fluctuations in the distance measurement between the embeddings and the cluster centers. Continuous or soft masks, however, are more sensitive to minor changes in the embedding positions. Equation 9 only holds the assumption that the soft clustering algorithm will generate the target masks after 1 iteration, at which the algorithm might not necessary to converge. This weaker assumption gives much higher flexibility on the embedding positions and leads to better generalization. In section 3.3 we will show that equation 10 leads to much worse performance than using equation 9.

2.4. Comparison with other methods

Comparing with previous methods towards stabilizing the clustering process in inference phase mentioned in section 1, the proposed method does not require any specific design of the objective function and remains flexible in both hard and soft mask estimation branches. However, it's harder for the proposed method to be purely end-to-end, since the hard mask estimation process is always necessary. One possible way is to unfold the hard clustering process from a random initialization to serve as layers in the network [20] so that a post-network clustering is not necessary.

3. EXPERIMENTS

3.1. Dataset

We evaluated our system on the public available WSJ0-2mix dataset for two speaker separation [5]. 30 hours of training and 10 hours of validation data are generated from speakers in si_tr_s from the datasets. The speech mixtures are generated by randomly selecting utterances from different speakers in the Wall Street Journal dataset (WSJ0) and mixing them at random relative signal-to-noise ratios (SNR) between -5 dB and 5 dB. A 5-hour evaluation set is generated in the same way using utterances from 16 unseen speakers in si_dt_05 and si_et_05. All the waveforms are resampled at 8 kHz. The input feature is the log spectral computed using short-time Fourier transform with 32 ms window length, 8 ms hop size, and the square root of hanning window. Wiener-filter like mask [15] is used as the target for soft mask estimation.

3.2. Model configuration

All models contain 4 Bi-directional LSTM layers with 300 hidden units in each direction followed by a fully-connected (FC) layer with $Tanh(\cdot)$ as nonlinearity function. The embedding dimension D is set to 20, resulting in 2580 hidden units (20×129) in the FC layer. The input features are splitted into non-overlapping chunks of 100-frame length before being fed into the networks. No regularizations or other training tricks are applied. Adam [21] is used as the optimizer with learning rate initialized to $1e^{-3}$. The learning rate is halved if no best model is found in the training set for 3 consecutive epochs. The maximum number of epochs was set to be 100. We report scale-invariant source-to-noise ratio (SI-SNR) [6, 9] as the evaluation metric.

3.3. Results and discussion

We first evaluate the proposed method on hard mask estimation task. K-means and spherical K-means are selected as the hard clustering algorithms. For spherical K-means, all embeddings are normalized to unit norm before processing. The networks are trained with equation 2 as objective. Two types of inference-phase strategies are tested: performing the same clustering algorithm from scratch, or using IBM as initialization and run 1 or 2 EM iterations depending on how the model is trained. The first strategy reflects the actual inference-phase performance, and the second strategy matches the training process of the network and reflects how well it has been trained (i.e. the upper-bound performance of inference phase). Table 1 shows the results. We can see that applying K-means from

scratch leads to only slightly worse performance than the oracle setting, showing that the network is behaving as how it is trained to be in the inference phase. Spherical K-means has a larger performance gap than K-means with higher actual and upper-bound performance. This might be due to the selection of α in equation 7. Regarding the convergence issues with multiple iterations in hard clustering mentioned in section 2.2.3, we can observe that both the actual and oracle performance of 2 EM models are worse than those with only 1 EM iteration, indicating that multiple iterations in hard clustering algorithms is not beneficial.

Table 1. Performance of hard clustering algorithms with 1 or 2 EM steps during training phase. Methods with * correspond to the ones trained with 2 EM steps.

Method	Inference	SI-SNRi (dB)
K-means	From scratch	8.8
	IBM, 1EM	9.1
K-means*	From scratch	8.8
	IBM, 2EM	9.0
Spherical K-means	From scratch	**9.2**
	IBM, 1EM	**10.0**
Spherical K-means*	From scratch	9.1
	IBM, 2EM	9.9

Table 2 provides the results of soft mask estimation. Two types of soft clustering algorithms, GMM and von Mises-Fisher distributions (vMF) [16], are selected together with K-means and spherical K-means respectively. Similarly, two types of inference-phase strategies can be applied depending on whether the estimated or oracle binary masks are used to initialize the soft clustering branch ('Hard' and 'IBM' in the table respectively). We can observe that K-means+GMM leads to significantly better performance comparing with the K-means hard clustering model, while minor improvement is achieved in the spherical K-means+vMF case. This indicates that clustering on unit hypersphere requires more investigation. Moreover, we find that models with 2 EM iterations are constantly worse than those with only 1 EM iterations, which matches the observation in hard mask estimation.

To show that equation 10 cannot guarantee that the target soft mask is the local minimum, we train a GMM model directly using equation 10 and run standard GMM EM iterations from scratch in inference phase until convergence. As shown in row 3 in table 2, the performance is significantly worse than any other models. This proves that it's not feasible to treat soft masks as local minimum.

We also investigate different combinations of hard and soft clustering algorithms that apply on different scales. For a same set of embeddings, the normalized ones are used for spherical K-means or vMF, and the original ones are used for K-means or GMM. The results are shown in the last two rows in table 2. Interestingly, combining K-means and vMF leads to complete failure in the estimation of hard masks. Looking into the distribution of the embeddings, we speculate that the failure is caused by the contrast between the two branches of the objectives, by which the model generates embeddings that favor vMF more. Possible resolutions might be either to initialize the model with a pre-trained K-means hard clustering model, or to assign a larger weight on the K-means objective in the multi-task learning objective.

Finally we look into the combination of a non-iterative soft clustering algorithm, the deep attractor network (DANet) [8, 9], together with a K-means branch for binary mask estimation. The results are shown in table 3. The original configuration for DANet is to firsts use IBM for a M-step in K-means to estimate cluster centers, and then use Softmax function on the dot product between embeddings and centers to generate soft masks. The test time estimation is done by use K-means to either estimate the binary mask or the cluster center and then perform the soft mask estimation. We find that the original DANet has huge gap between training- and inference-phases due to the inaccurate estimation by K-means. After adding the K-means branch, the actual performance significantly improves and achieves the highest result among all systems. This indicates that the proposed method can be applied to either iterative or non-iterative methods for a performance improvement in soft mask estimation.

Table 2. Performance of soft clustering algorithms with 1 or 2 EM steps during training phase. Methods with * correspond to the ones trained with 2 EM steps.

Method	Inference	SI-SNRi (dB)
K-means+GMM	Hard, 1EM	9.3
	IBM, 1EM	10.1
K-means+GMM*	Hard, 2EM	9.3
	IBM, 2EM	9.9
GMM	From scratch	5.9
Spherical K-means+vMF	Hard, 1EM	**9.4**
	IBM, 1EM	10.0
Spherical K-means+vMF*	Hard, 2EM	9.1
	IBM, 2EM	9.7
K-means+vMF	Hard, 1EM	-8.7
	IBM, 1EM	9.8
Spherical K-means+GMM	Hard, 1EM	9.2
	IBM, 1EM	**10.2**

Table 3. Performance of DANet with different configurations.

Method	Inference	SI-SNRi (dB)
DANet	Hard centroid	8.5
	Hard mask	8.5
	IBM	9.8
DANet* [8, 9]	Hard centroid	9.6
K-means+DANet	Hard	**9.7**
	IBM	**9.9**

4. CONCLUSION

We propose a simple method for stabilizing time-frequency mask estimation in clustering-based source separation systems. The key concept is to train the embedding generation system so that ideal binary mask is a local minimum/maximum for a selected hard clustering algorithm. Soft clustering algorithms can then benefit from the hard clustering algorithm by using the estimated binary mask as the initialization. Experiments show that the proposed method is robust across various clustering methods and significantly bridges the performance gap between training- and inference-phases.

5. ACKNOWLEDGEMENT

This work was funded by a grant from National Institute of Health, NIDCD, DC014279, National Science Foundation CAREER Award, and the Pew Charitable Trusts.

6. REFERENCES

[1] Hiroshi Sawada, Shoko Araki, and Shoji Makino, "Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 3, pp. 516–527, 2011.

[2] Daniel Patrick Whittlesey Ellis, *Prediction-driven computational auditory scene analysis*, Ph.D. thesis, Massachusetts Institute of Technology, 1996.

[3] DeLiang Wang and Guy J Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*, Wiley-IEEE press, 2006.

[4] Ke Hu and DeLiang Wang, "An unsupervised approach to cochannel speech separation," *IEEE Transactions on audio, speech, and language processing*, vol. 21, no. 1, pp. 122–131, 2013.

[5] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 31–35.

[6] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey, "Single-channel multi-speaker separation using deep clustering," *arXiv preprint arXiv:1607.02173*, 2016.

[7] Yi Luo, Zhuo Chen, John R Hershey, Jonathan Le Roux, and Nima Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 61–65.

[8] Zhuo Chen, Yi Luo, and Nima Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 246–250.

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

[10] Zhong-Qiu Wang, Jonathan Le Roux, and John R Hershey, "Alternative objective functions for deep clustering," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[11] Lukas Drude and Reinhold Haeb-Umbach, "Tight integration of spatial and spectral features for bss with deep clustering embeddings," in *Proc. Interspeech*, 2017, pp. 2650–2654.

[12] Zhuo Chen, Jinyu Li, Xiong Xiao, Takuya Yoshioka, Huaming Wang, Zhenghao Wang, and Yifan Gong, "Cracking the cocktail party problem by multi-beam deep attractor network," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 437–444.

[13] Yuzhou Liu and DeLiang Wang, "A casa approach to deep learning based speaker-independent co-channel speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5399–5403.

[14] DeLiang Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech separation by humans and machines*, pp. 181–197. Springer, 2005.

[15] Hakan Erdogan, John R Hershey, Shinji Watanabe, and Jonathan Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 708–712.

[16] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra, "Clustering on the unit hypersphere using von mises-fisher distributions," *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1345–1382, 2005.

[17] Eric Jang, Shixiang Gu, and Ben Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[18] Yoshua Bengio, Nicholas Léonard, and Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[19] Felix Weninger, John R Hershey, Jonathan Le Roux, and Björn Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proceedings 2nd IEEE Global Conference on Signal and Information Processing, GlobalSIP, Machine Learning Applications in Speech Processing Symposium, Atlanta, GA, USA*, 2014.

[20] John R Hershey, Jonathan Le Roux, and Felix Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[21] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.