# ALTERNATIVE OBJECTIVE FUNCTIONS FOR DEEP CLUSTERING

*Zhong-Qiu Wang*[1,2]*, Jonathan Le Roux*[1]*, John R. Hershey*[1]

[1]Mitsubishi Electric Research Laboratories (MERL), USA
[2]Department of Computer Science and Engineering, The Ohio State University, USA

## ABSTRACT

The recently proposed deep clustering framework represents a significant step towards solving the cocktail party problem. This study proposes and compares a variety of alternative objective functions for training deep clustering networks. In addition, whereas the original deep clustering work relied on k-means clustering for test-time inference, here we investigate inference methods that are matched to the training objective. Furthermore, we explore the use of an improved *chimera network* architecture for speech separation, which combines deep clustering with mask-inference networks in a multi-objective training scheme. The deep clustering loss acts as a regularizer while training the end-to-end mask inference network for best separation. With further iterative phase reconstruction, our best proposed method achieves a state-of-the-art 11.5 dB signal-to-distortion ratio (SDR) result on the publicly available wsj0-2mix dataset, with a much simpler architecture than the previous best approach.

***Index Terms***— deep clustering, speaker-independent multi-talker speech separation, chimera network, cocktail party problem

## 1. INTRODUCTION

Recent advances have been made on the notoriously hard single-channel speaker-independent multi-talker speech separation problem, *a.k.a.* the cocktail party problem, using a framework known as *deep clustering*. The deep clustering framework [1, 2] projects each time-frequency (T-F) unit to a high-dimensional embedding such that the pairs of embeddings dominated by the same speaker are closer to each other while those dominated by different speakers are farther apart. This way, speaker assignment can be determined by running a simple clustering on the embeddings at test time. The deep clustering approach was the first to address the *permutation problem*, where the correspondence between the outputs of an algorithm and the references is an arbitrary permutation. This paper follows similar principles but explores alternative formulations of the training and inference objectives, both alone and in combination with other strategies.

In [2], direct optimization of the separation performance was explored using end-to-end training through the clustering step using a segment-level permutation-free objective function that considers all possible permutations of the references when comparing them with the outputs for the current speech segment (typically 400 frames). Only the error for the best matching permutation is used in training. A key property of this approach is that the same networks can be used with any number of sources.

An alternative approach is to perform *direct mask inference* using the permutation-free objective function with networks that directly estimate the labels for a fixed number of sources. Direct mask inference was first used in [1] as a baseline method, using a combination of long short term memory (LSTM) recurrent neural networks (RNNs), and a segment-level permutation-free objective, but without showing good performance. The model in [2] is also a form of direct mask inference, since it uses the segment-level permutation-free objective instead of the deep clustering network, although the underlying architecture is inspired by deep clustering. The direct mask inference approach was revisited in [3] using DNNs and frame-level permutation-free training, along with an oracle "tracing" method to resolve permutations between frames. In [4], the combination of LSTMs and segment-level permutation-free training was adopted, where it was shown to perform nearly as well as deep clustering.

The two approaches appear to be complementary, however, in the sense that combining them into a chimera network, as in [5], can produce better results than either on its own. In the chimera network, the deep clustering task functions as a regularizer to guide the mask inference to perform better separation.

Together with the strong learning power of deep neural networks, all of these approaches have demonstrated overwhelming advantages over previous approaches including graphical modeling approaches [6], spectral clustering approaches [7], and CASA methods [8]

In this study, we introduce several alternative deep clustering objectives that aim to improve clustering performance. We also introduce energy-dependent weighting to reduce the influence of low-energy bins. For each objective function, an additional run-time inference algorithm is proposed to obtain a ratio mask for separation. For mask-inference networks, we show improvements using an $L_1$ loss and logistic sigmoid activation in the output layer when estimating the phase-sensitive mask [9]. We show that these improvements extend to the combination with deep clustering via an improved chimera framework.

## 2. DEEP CLUSTERING

The key idea of deep clustering is to use a powerful neural network to learn a high-dimensional embedding for each T-F unit such that the embeddings belonging to the same speaker are close to each other in the embedding space, and farther otherwise. This way, simple clustering methods such as k-means can be performed on the learned embeddings to perform separation at the test stage.

The network computes a unit-length embedding vector $v_i \in \mathbb{R}^{1\times D}$ corresponding to the $i$-th time-frequency element, which corresponds to a particular pair of time-frequency indices $t$ and $f$. Likewise, $y_i \in \mathbb{R}^{1\times C}$ is a one-hot label vector indicating which source in a mixture dominates time-frequency bin $i$. Vertically stacking these, we form the embedding matrix $V \in \mathbb{R}^{TF\times D}$, and the label matrix $Y \in \mathbb{R}^{TF\times C}$. The embeddings are learned in a way such that the affinity matrix can be approximated from the embeddings

This work was done while Z.-Q. Wang was an intern at MERL.

by minimizing the following objective function:

$$\begin{aligned}\mathcal{L}_{\text{DC,classic}}(V,Y) &= \|VV^{\mathsf{T}} - YY^{\mathsf{T}}\|_{\mathrm{F}}^2 \\ &= \|V^{\mathsf{T}}V\|_{\mathrm{F}}^2 + \|Y^{\mathsf{T}}Y\|_{\mathrm{F}}^2 - 2\|V^{\mathsf{T}}Y\|_{\mathrm{F}}^2 \\ &= \sum_{i,j}[\langle v_i, v_j\rangle - \langle y_i, y_j\rangle]^2 \end{aligned} \quad (1)$$

The network architecture of deep clustering is shown in Fig. 1(a).

## 3. ALTERNATIVE OBJECTIVE FUNCTIONS

**Graph Laplacian distance:** A problem in the original deep clustering formula when used to infer $Y$ is that it contains the term $\|YY^{\mathsf{T}}\|_F^2 = \sum_c N_c^2$, where $N_c = \sum_i y_{i,c}$ is the number of bins dominated by speaker $c$. This regularizes the solution toward clusters of equal size. To avoid this, we can normalize $YY^{\mathsf{T}}$ and $VV^{\mathsf{T}}$ so the objective is less dependent on cluster size. Consider the following objective, where $D_{\mathsf{V}} = \mathrm{diag}(VV^{\mathsf{T}}\mathbf{1})$ and similarly for $D_{\mathsf{Y}}$:

$$\begin{aligned}\mathcal{L}_{\text{DC,L}}(V,Y) &= \|D_{\mathsf{V}}^{-\frac{1}{2}}VV^{\mathsf{T}}D_{\mathsf{V}}^{-\frac{1}{2}} - D_{\mathsf{Y}}^{-\frac{1}{2}}YY^{\mathsf{T}}D_{\mathsf{Y}}^{-\frac{1}{2}}\|_{\mathrm{F}}^2 \\ &= \|V^{\mathsf{T}}D_{\mathsf{V}}^{-1}V\|_{\mathrm{F}}^2 + C - 2\,\|V^{\mathsf{T}}D_{\mathsf{V}}^{-\frac{1}{2}}D_{\mathsf{Y}}^{-\frac{1}{2}}Y\|_{\mathrm{F}}^2. \end{aligned} \quad (2)$$

If $V$ were a partition matrix, this would be the *chi-squared distance* between $V$ and $Y$ [7, 10]. This form also arises in spectral clustering, where the affinity matrix is interpreted as a weighted graph. The matrix $L = I - D_{\mathsf{V}}^{-\frac{1}{2}}VV^{\mathsf{T}}D_{\mathsf{V}}^{-\frac{1}{2}}$ is a normalized graph Laplacian for the affinity matrix $VV^{\mathsf{T}}$, and its eigendecomposition optimizes the *normalized cut* criterion for graph partitioning [7, 10]. $\mathcal{L}_{\text{DC,L}}$ is the difference between the graph Laplacian of $VV^{\mathsf{T}}$ and that of the ideal affinity matrix $YY^{\mathsf{T}}$, which may help in balancing the classes according to the normalized cut criterion.

**Stochastic normalization:** Another normalization comes from viewing the affinities as a stochastic matrix, leading to the random-walk interpretation of spectral clustering [11]. This motivates normalizing the symmetric affinity matrix to be doubly stochastic. For an ordinary affinity matrix this would require an *iterative scaling* procedure [12]; however, because the deep clustering affinity matrix is a symmetric product, we can accomplish this in closed form by conditioning $V$:

$$\bar{V} = \mathrm{diag}(V1)^{-1}V, \qquad \check{V} = \bar{V}\,\mathrm{diag}(1^{\mathsf{T}}\bar{V})^{-1/2}, \quad (3)$$

so that $\check{V}\check{V}^{\mathsf{T}}$ is doubly stochastic [13]. With $\check{Y}$ similarly obtained from $Y$, the doubly stochastic objective is:

$$\mathcal{L}_{\text{DC,S}}(V,Y) = \|\check{V}\check{V}^{\mathsf{T}} - \check{Y}\check{Y}^{\mathsf{T}}\|_{\mathrm{F}}^2. \quad (4)$$

**Deep LDA:** In [1], k-means is used to cluster the embeddings to infer the class labels for each TF bin. However, the training objective (1), as a function of the embeddings $V$ given the reference labels $Y$, is different from the inference objective optimized by k-means, as a function of the inferred labels $Y$ given the embeddings $V$. Here we consider modifying the training objective so that it matches that of k-means algorithm used for inference. The k-means objective minimizes the within-class variance of the embeddings $V$ as a function of their class assignments $Y$. However, training the embeddings $V$ given reference $Y$ under the k-means objective leads to the trivial solution in which all embeddings are the same. Instead, we can use the ratio of the within-class variance to the total variance,

$$\begin{aligned}\mathcal{L}_{\text{DC,LDA}}(V,Y) &= \frac{\|V - Y(Y^{\mathsf{T}}Y)^{-1}Y^{\mathsf{T}}V\|_{\mathrm{F}}^2}{\|V - \mathbf{1}(\mathbf{1}^{\mathsf{T}}\mathbf{1})^{-1}\mathbf{1}^{\mathsf{T}}V\|_{\mathrm{F}}^2}, \\ &= \frac{\sum_{c=1}^{C}\sum_{i,y_{i,c}=1}(v_i - \bar{v}^{(c)})^2}{\sum_i (v_i - \bar{v})^2}, \end{aligned} \quad (5)$$

where $\bar{v}$ is the mean of all the embeddings and $\bar{v}^{(c)}$ the mean of the embeddings belonging to speaker $c$. This is the same objective optimized by linear discriminant analysis [14], except that $V$ is here produced by a neural network instead of being a linear function of the data points.

Note that when inferring $Y$ given $V$, the denominator is a constant, and we recover the k-means objective: training and inference objectives are matched.

**Whitened k-means:** We consider an alternative to the Deep LDA objective above, where rather than using the denominator in (5) to avoid a trivial solution, we instead normalize $V$ to have identity-covariance. That is we use a whitened $\tilde{V} = V(V^TV)^{-1/2}$ in the numerator of (5) to obtain:

$$\begin{aligned}\mathcal{L}_{\text{DC,W}}(V,Y) &= \|V(V^{\mathsf{T}}V)^{-\frac{1}{2}} - Y(Y^{\mathsf{T}}Y)^{-1}Y^{\mathsf{T}}V(V^{\mathsf{T}}V)^{-\frac{1}{2}}\|_{\mathrm{F}}^2 \\ &= D - \mathrm{tr}\big((V^{\mathsf{T}}V)^{-1}V^{\mathsf{T}}Y(Y^{\mathsf{T}}Y)^{-1}Y^{\mathsf{T}}V\big). \end{aligned} \quad (6)$$

Note that this objective function is a linear transform away from $\|V(V^{\mathsf{T}}V)^{-1}V^{\mathsf{T}} - Y(Y^{\mathsf{T}}Y)^{-1}Y^{\mathsf{T}}\|_{\mathrm{F}}^2$, which, similarly to (2), would be the chi-squared distance between $V$ and $Y$ if $V$ were a partition.

**Introducing weights:** Discarding or reducing the influence of T-F bins in silence regions is found to be very important for training deep clustering networks. The estimated mask value for such low-energy bins has little influence on the output, and their labelling is somewhat arbitrary. It is thus likely counterproductive to force the network to learn how to create embeddings for the many such bins. By filtering them out, the network can focus on learning embeddings for the T-F bins that actually contain some speech.

The weighting is applied via a diagonal weight matrix $W = \mathrm{diag}(w)$. For the classic deep clustering loss, the weighted loss function is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{DC,classic},W}(V,Y) &= \|W^{1/2}\left(VV^{\mathsf{T}} - YY^{\mathsf{T}}\right)W^{1/2}\|_{\mathrm{F}}^2 \\ &= \sum_{i,j} w_i w_j[\langle v_i, v_j\rangle - \langle y_i, y_j\rangle]^2. \end{aligned} \quad (7)$$

The weighting mechanism can be efficiently implemented by applying $\sqrt{w}$ to both $V$ and $Y$, and then using Eq. (1) to compute the loss. The weights can be introduced similarly for the other cost functions.

There are multiple ways to define the weights for training. A simple option is to use binary voice activity weights $W_{\text{VA}}$ to filter out the T-F bins where none of the sources are significantly active [1], where a source is considered active in T-F bins where its magnitude is within some threshold from its largest magnitude in the utterance: $w_i = \max_k \left[10\log_{10}(|s_{k,i}|^2/\max_j |s_{k,j}|^2) > \beta\right]$, where $[\cdot]$ is the Iverson bracket and $|s_{k,i}|$ represents the magnitude of the $i$-th T-F bin of the clean source $k$. Another option avoids a hard threshold hyper-parameter and uses soft weights. We use here magnitude ratio weights $W_{\text{MR}}$ defined as the ratio of the mixture magnitude at T-F bin $i$ over the sum of the mixture magnitudes at all bins within an utterance: $w_i = |x_i|/\sum_j |x_j|$, where $|x|$ is the magnitude of the mixture.

**Matched inference:** Whereas in the above we considered training objectives for the deep network embeddings $V$, we here consider inference objectives for the class assignments $Y$. In particular, we use the same objective for inference that was used in training in each case. To make this possible, we relax the discrete class assignments and use gradient descent to infer a continuous mask $\hat{Y}$ given the network's embeddings $\hat{V}$: $\hat{Y} = \arg\min_Y \mathcal{L}_{\text{DC}}(\hat{V}, Y)$, for the various training objectives. We parameterize the mask as $Y = \mathrm{logistic}(Z)$, where $Z$ are unconstrained real values, and optimize the above objective with respect to $Z$ using a gradient descent-based algorithm such as Adam. Note that the deep LDA training objective is de-

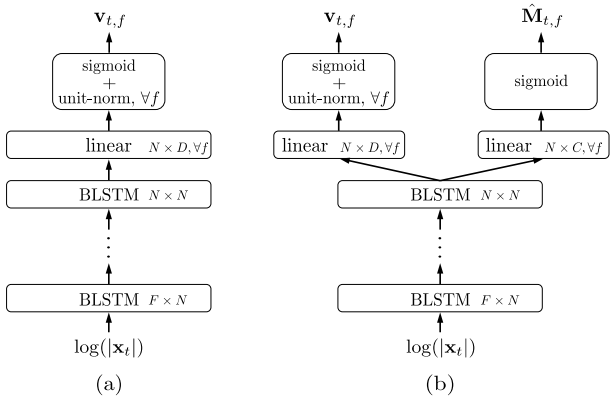


**Fig. 1**. (a) Deep clustering network, (b) Chimera++ network

signed to match the k-means inference objective, but we can still perform inference of continuous class assignments, using gradient descent instead of k-means, to further improve the result. We initialize the parameters using the binary mask obtained from k-means clustering. More specifically, we first modify all the ones in the binary mask to 0.99 and all the zeros to 0.01, and then apply the logit function to get the pre-activation values of $Z$ for initialization. Note that for inference, the parameters of the network and $\hat{V}$ stay fixed, so the optimization is efficient.

## 4. ARCHITECTURE IMPROVEMENTS

**Improved mask-inference networks:** In the original deep clustering paper [1], as a comparison with deep clustering, the authors proposed to train a conventional mask-inference (MI) network using a segment-level permutation-free objective based on the magnitude spectrum approximation (MSA):

$$\mathcal{L}_{\text{MI,MSA}} = \min_{\pi \in \mathcal{P}} \sum_{c} \|\hat{M}_c \circ |X| - |S_{\pi(c)}|\|^2, \tag{8}$$

where $\mathcal{P}$ is the set of permutations on $\{1, \ldots, C\}$, $|X|$ the mixture magnitude, $\hat{M}_c$ the $c$-th estimated mask, and $|S_c|$ the magnitude of the $c$-th reference source. Although this baseline performed poorly, in a follow-up paper [2], this permutation-free objective was used to train an MSA estimation network built upon a deep clustering network with unfolded k-means steps, this time leading to state-of-the-art results. In [9] the phase-sensitive spectrum approximation (PSA) was shown to outperform MSA for separating speech from non-stationary interference. Subsequently in [4], PSA estimation was combined with the segment-level permutation-free objective resulting in improved performance. It is common in PSA to truncate the mask values to the range $[0, 1]$, and therefore truncate the phase-sensitive spectrum target to the range $[0, |X|]$. Using $\mathrm{T}_a^b(x) = \min(\max(x, a), b)$, the truncated PSA (tPSA) objective is

$$\mathcal{L}_{\text{MI,tPSA},L_2} = \min_{\pi \in \mathcal{P}} \sum_{c} \Big\| \hat{M}_c \circ |X| - \mathrm{T}_0^{|X|}\left(|S_{\pi(c)}| \circ \cos(\theta_X - \theta_{\pi(c)})\right) \Big\|^2, \tag{9}$$

where $\theta_X$ is the mixture phase and $\theta_c$ the phase of the $c$-th source.
To allow the network more flexibility, we use the logistic sigmoid activation, whereas the softmax activation was used in [4]. We also introduce a loss function that uses the $L_1$ distance instead of the squared $L_2$ distance in Eq. (9), denoted by $\mathcal{L}_{\text{MI,PSA},L_1}$. This is motivated by the fact that the histogram of clean STFT magnitudes in anechoic environments is similar to a Laplacian distribution (or a super-Gaussian distribution [15]), and so is the error term distribution, as suggested in our previous study [16]. In our experiments, we found that using the logistic sigmoid activation in combination with the $\mathcal{L}_{\text{MI,PSA},L_1}$ loss leads to better performance than corresponding results in [4].
**Chimera++ network:** In [5], a chimera network is introduced that combines deep clustering with MI in a multi-task learning fashion, leveraging the regularizing property of the deep clustering loss and the simplicity of the mask-inference network. In the original chimera network, the mask inference branch grows out from the embedding layer. The motivation for this is unclear as the embedding layer exists to satisfy the deep clustering objective, whereas mask inference does not require such tight coupling with the embeddings. We propose to graft the mask inference at the BLSTM hidden layer output, yielding a conceptually simpler and computationally faster network, shown in Fig. 1(b). We refer to it as chimera++. The loss function we minimize is a weighted sum of a deep clustering loss and an MI loss:

$$\mathcal{L}_{\text{chi}_\alpha^{++}} = \alpha \mathcal{L}_{\text{DC}}(V, Y) + (1 - \alpha)\mathcal{L}_{\text{MI}}. \tag{10}$$

At run time, we only need the MI output to make predictions.

## 5. EXPERIMENTAL VALIDATION

### 5.1. Setup

We evaluate our algorithms on the publicly-available wsj0-2mix dataset [1], which has been used by many studies after the debut of the deep clustering algorithm. It contains 20,000 utterances ($\sim$30h) in the training data and 5,000 utterances ($\sim$10h) in the validation data, both of which are created by randomly mixing two utterances of two randomly-chosen speakers from the WSJ0 training data (si_tr_s). Each mixture of the 3,000 utterances ($\sim$5h) in the testing data is generated by mixing two utterances from two randomly-chosen speakers in the WSJ0 validation (si_dt_05) and testing set (si_et_05). Note that the speakers in our validation set are also included in the training set (of course, the utterances are different), so we denote it as closed speaker condition (CSC), while the test set consists of an unseen set of speakers, therefore we denote it as open speaker condition (OSC). The SNR of each mixture is randomly drawn between 0 dB and 10 dB. The sampling rate is 8 kHz.
Our model contains four BLSTM layers each with 600 units in each direction. The network is trained from scratch on 400-frame segments using the Adam algorithm. 0.3 dropout is applied on the output of each BLSTM layer except the last one. No curriculum learning or recurrent dropout [2, 17, 18] is incorporated in our system. The window length is 32ms, the hop size is 8ms, and the square root of the Hann window is used as the analysis window. 256-point DFT is performed to extract the 129-dimensional log magnitude feature of each frame for BLSTM training. The run-time clustering is always performed on the entire utterance. Following [2], $\beta$ in $W_{\text{VA}}$ is empirically set to -40. For multi-task learning, although the model is trained using a combination of deep clustering loss and MI loss, we only use the MI loss during validation for model selection. At run time, we use the output from the MI branch as the masks for separation. The $\alpha$ coefficient is set through cross-validation.

### 5.2. Iterative Phase Reconstruction

In most deep learning studies for speech separation or enhancement, only the magnitude is enhanced, and the noisy phase is used directly for time-domain re-synthesis, largely because the phase pat-

**Table 1**. SDR (dB) performance on wsj0-2mix.

| Approaches | Dropout | CSC | OSC |
|---|---|---|---|
| DC (classic, equal weights) | 0.0 | 9.6 | 9.5 |
| DC (classic, $W_{\text{VA}}$) | 0.0 | 9.8 | 9.7 |
| DC (classic, $W_{\text{VA}}$) | 0.3 | 10.0 | 9.9 |
| DC (classic, $W_{\text{MR}}$) | 0.0 | 10.0 | 10.0 |
| DC (classic, $W_{\text{MR}}$) | 0.3 | 10.3 | 10.2 |
| +run-time inference | - | 10.7 | 10.7 |
| DC (LDA, $W_{\text{VA}}$) | 0.0 | 9.5 | 9.4 |
| DC (LDA, $W_{\text{VA}}$) | 0.3 | 9.8 | 9.7 |
| + run-time inference | - | 10.2 | 10.1 |
| DC (S, $W_{\text{VA}}$) | 0.3 | 10.1 | 10.0 |
| DC (S, $W_{\text{MR}}$) | 0.3 | 10.2 | 10.1 |
| + run-time inference | - | 10.7 | 10.7 |
| DC (L, $W_{\text{VA}}$) | 0.3 | 10.2 | 10.2 |
| + run-time inference | - | 10.7 | 10.7 |
| DC (W, $W_{\text{VA}}$) | 0.3 | 10.3 | 10.3 |
| DC (W, $W_{\text{MR}}$) | 0.3 | 10.4 | 10.4 |
| + run-time inference | - | 10.9 | 10.9 |
| MI (softmax, tPSA, $L_2$) | 0.3 | 9.1 | 9.1 |
| MI (softmax, tPSA, $L_1$) | 0.3 | 10.0 | 9.8 |
| MI (sigmoid, tPSA, $L_1$) | 0.3 | 10.1 | 10.0 |
| chimera++: | | | |
| DC (W, $W_{\text{VA}}$)+MI (sigmoid, tPSA, $L_1$) | 0.3 | 10.9 | 10.9 |
| DC (W, $W_{\text{MR}}$)+MI (softmax, tPSA, $L_1$) | 0.3 | 11.1 | 11.1 |
| DC (W, $W_{\text{MR}}$)+MI (sigmoid, tPSA, $L_1$) | 0.3 | 11.1 | 11.2 |
| + Griffin-Lim | - | 11.2 | 11.3 |
| + MISI | - | 11.4 | 11.5 |

**Table 2**. SDR (dB) comparison with other systems on wsj0-2mix.

| Approaches | CSC | OSC |
|---|---|---|
| DC [1, 2] | - | 10.8 |
| DANet-6 anchor [17, 18] | - | 10.4 |
| uPIT (relu, PSA, $L_2$) + stacking [4] | 10.0 | 10.0 |
| Proposed | **11.4** | **11.5** |
| Oracle Masks: | | |
| Magnitude Ratio Mask | 12.5 | 12.7 |
| Ideal Binary Mask | 13.2 | 13.5 |
| PSA Mask | 16.2 | 16.4 |

tern is very random and therefore hard to enhance. Conventionally, iterative methods, such as the Griffin-Lim algorithm [19], can recover the clean phase to some extent starting from the noisy phase and a good estimated magnitude by iteratively performing STFT and iSTFT [20]. There are some previous attempts at applying Griffin-Lim to deep-learning-based speech enhancement [21, 22, 23]. However, Griffin-Lim performs iterative reconstruction for each source independently and fails to exploit the constraint that we here separate multiple sources that should sum up to the mixture. We therefore propose to jointly reconstruct the phase of all sources starting from their estimated magnitudes and the noisy phase via the multiple input spectrogram inversion (MISI) algorithm [24], where the sum of the reconstructed time-domain signals after each iteration is constrained to be equal to the mixture signal. The iteration number in MISI and Griffin-Lim is set to five in our experiments.

### 5.3. Results

The SDR results of our algorithm on the wsj0-2mix dataset are reported in Table 1. When using the classic deep clustering loss together with weights $W_{\text{VA}}$ and 0.3 feed-forward dropout, we only get to 9.9 dB, which is 0.4 dB worse than the 10.3 dB SDR result reported in [2], likely because the recurrent dropout and curriculum learning are not included in our system. Using weights $W_{\text{MR}}$ for training decreases the gap to 0.1 dB (10.2 vs. 10.3 dB), indicating the effectiveness of the weighting mechanism in the deep clustering algorithm. The LDA objective function gives an unimpressive 9.7 dB SDR performance, possibly because it does not directly compare every T-F pairs as the other objective functions. The best performance among the various deep clustering objective functions is obtained by the $L_{\text{DC,W}}$ objective function, with 10.4 dB. Future work should aim to understand why $L_{\text{DC,W}}$ outperforms the other objectives in this task and whether this difference holds in other cases.

Starting from the estimated binary masks, run-time inference consistently improves the performance for all the objective functions, leading to 10.9 dB when used in combination with $L_{\text{DC,W}}$. This is already 0.1 dB better than the previous state-of-the-art result [2].

We managed to reproduce the key SDR results in [4] using the MI network. Note that in [4], even with second-stage stacking, the MI network can only reach 10.0 dB, and 9.4 dB without stacking. In our system, even without further stacking, we obtained a comparable 10.0 dB SDR by using the logistic sigmoid activation together with the $L_1$ loss for the PSA objective. Switching to the $L_1$ loss gives a 0.8 dB improvement by itself, from 9.0 to 9.8 dB. This justifies the benefits of the $L_1$ loss for the PSA objective.By replacing the softmax activation with the logistic sigmoid activation, we observe another 0.2 dB improvement (from 9.8 to 10.0 dB).

Combining deep clustering with MI via multi-task learning improves the performance significantly to 11.2 dB. Note that the network is trained starting from random initialization. In contrast, training the same MI network alone only reaches 10.0 dB, while training the deep clustering network alone only gets to 10.4 dB. This improvement is likely due to the regularizing effect of the deep clustering loss. In our experiments, using weights $W_{\text{MR}}$ in the deep clustering branch significantly improves the performance from 10.9 to 11.2 dB, and using sigmoid activation in the MI output is also better than softmax (11.2 vs. 11.1 dB). Since we only need to use the MI output at run time, the system is also faster as it does not need to produce embeddings and perform run-time clustering anymore.

With the mixture phase and magnitudes estimated from the MI branch, we perform iterative phase reconstruction using MISI, pushing the performance further to 11.5 dB SDR, our current best result. In constrast, using Griffin-Lim independently on each source only reaches 11.3 dB. As the FFT algorithm is very fast even on CPU, we think that iterative phase reconstruction is feasible in practice.

Table 2 lists the performance of competitive approaches on the same wsj0-2mix dataset, together with the performance of various oracle masks. The ideal binary mask is computed based on which source is dominant at each T-F unit and the magnitude ratio mask is computed using the magnitude of each source over the sum of all the magnitudes at each T-F unit. Our result is 0.7 dB better than the previous state-of-the-art algorithm by [2].

## 6. CONCLUDING REMARKS

We proposed multiple alternative loss functions for training deep clustering networks. Among them, the $L_{\text{DC,W}}$ loss leads to the best performance. Run-time inference can lead to consistently better performance for all deep clustering loss functions, at the price of an increase in computational cost. Combining MI with deep clustering in our improved chimera++ architecture significantly improves MI performance. Finally, we show that our best architecture is able to output magnitudes with sufficient quality for a phase reconstruction algorithm such as MISI to further improve performance.

## 7. REFERENCES


[1] 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.

[2] 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.

[3] 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 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017.

[4] 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, 2017.

[5] 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 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017.

[6] J. R. Hershey, S. Rennie, P. A. Olsen, and T. T. Kristjansson, "Super-Human Multi-Talker Speech Recognition: A Graphical Modeling Approach," *Computer Speech & Language*, vol. 24, no. 1, 2010.

[7] F. Bach and M. Jordan, "Learning Spectral Clustering, with Application to Speech Separation," *The Journal of Machine Learning Research*, vol. 7, 2006.

[8] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Hoboken, NJ: Wiley-IEEE Press, Sep. 2006.

[9] 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 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015.

[10] M. Meilă, "Local Equivalences of Distances Between ClusteringsA Geometric Perspective," *Machine Learning*, vol. 86, no. 3, 2012.

[11] M. Meilă and J. Shi, "Learning Segmentation by Random Walks," in *Advances in neural information processing systems*, 2001.

[12] O. Pretzel, "Convergence of the iterative scaling procedure for non-negative matrices," *Journal of the London Mathematical Society*, vol. 2, no. 2, 1980.

[13] Z. Yang, J. Corander, and E. Oja, "Low-rank doubly stochastic matrix decomposition for cluster analysis," *Journal of Machine Learning Research*, vol. 17, no. 187, 2016.

[14] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[15] K. Kumatani, A. Takayuki, K. Yamamoto, J. McDonough, B. Raj, R. Singh, and I. Tashev, "Microphone Array Processing for Distant Speech Recognition: Towards Real-World Deployment," in *Annual Summit and Conference in Signal & Information Processing*, 2012.

[16] Z.-Q. Wang and D. Wang, "Recurrent Deep Stacking Networks for Supervised Speech Separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017.

[17] Z. Chen, Y. Luo, and N. Mesgarani, "Deep Attractor Network for Single-Microphone Speaker Separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017.

[18] ——, "Speaker-Independent Speech Separation with Deep Attractor Network," *arXiv preprint arXiv:1707.03634*, Jul 2017.

[19] D. W. Griffin and J. S. Lim, "Signal Estimation from Modified Short-Time Fourier Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, 1984.

[20] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux, "Phase Processing for Single-Channel Speech Enhancement: History and Recent Advances," *IEEE Signal Processing Magazine*, vol. 32, no. 2, 2015.

[21] K. Han, Y. Wang, D. Wang, W. S. Woods, and I. Merks, "Learning Spectral Mapping for Speech Dereverberation and Denoising," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 6, 2015.

[22] Y. Zhao, Z.-Q. Wang, and D. Wang, "A Two-Stage Algorithm for Noisy and Reverberant Speech Enhancement," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017.

[23] K. Li, B. Wu, and C.-H. Lee, "An Iterative Phase Recovery Framework with Phase Mask for Spectral Mapping with an Application to Speech Enhancement," in *Proc. Interspeech*, Sep. 2016.

[24] D. Gunawan and D. Sen, "Iterative Phase Estimation for the Synthesis of Separated Sources from Single-Channel Mixtures," in *IEEE Signal Processing Letters*, 2010.