Discriminatively Trained Recurrent Neural Networks for Single-Channel Speech Separation

Felix Weninger*, John R. Hershey†, Jonathan Le Roux†, Björn Schuller*
*Machine Intelligence & Signal Processing Group (MISP), Technische Universität München, 80290 Munich, Germany
Email: {weninger,schuller}@tum.de
†Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA
Email: {hershey,leroux}@merl.com

***Abstract*—This paper describes an in-depth investigation of training criteria, network architectures and feature representations for regression-based single-channel speech separation with deep neural networks (DNNs). We use a generic discriminative training criterion corresponding to optimal source reconstruction from time-frequency masks, and introduce its application to speech separation in a reduced feature space (Mel-domain). A comparative evaluation of time-frequency mask estimation by DNNs, recurrent DNNs and non-negative matrix factorization on the 2nd CHiME Speech Separation and Recognition Challenge shows consistent improvements by discriminative training, whereas Long Short-Term Memory recurrent DNNs obtain the overall best results. Furthermore, our results confirm the importance of fine-tuning the feature representation for DNN training.**

***Index Terms*—speech enhancement; deep neural networks; discriminative training**

I. Introduction

Single-channel source separation aims to recover one or more source signals of interest from a mixture of signals. An important application in audio signal processing is to obtain clean speech signals from single-channel recordings with non-stationary noises, in order to facilitate human-human or human-machine communication in unfavorable acoustic environments. Popular algorithms for this task include model-based approaches such as non-negative matrix factorization (NMF) [1]–[3] and more recently, supervised learning of time-frequency masks for the noisy spectrum [4]–[7]. However, it is notable that these methods do not directly optimize the actual objective of source separation, which is an optimal reconstruction of the desired signal(s). Initial studies have recently shown the benefit of incorporating such criteria for NMF [8] and deep neural network [9] based speech separation.

In this paper, we consolidate earlier work on discriminative speech separation by starting from a generic discriminative training objective for optimizing SNR. We then use this framework to derive a novel discriminative objective for mask estimation in a reduced feature space (here, the Mel-domain) from which a full-resolution result is obtained by filtering. Furthermore, we show the importance of feature and training target representation in combination with deep learning techniques for single-channel speech separation. Finally, by investigating discriminative training of Long Short-Term Memory recurrent neural networks for speech separation, we show that good design of discriminative objective functions is complementary to improved recurrent neural network architectures circumventing the vanishing gradient problem.

II. Speech separation by time-frequency filtering

The problem of single-channel speech separation is to obtain an estimate $\hat{s}(t)$ of a target speech signal $s(t)$ from a mixture signal $m(t)$, which also contains background noise $n(t)$. A popular approach is to work in the time-frequency domain, for example obtained by short-time Fourier transform (STFT) based on a discrete Fourier transform (DFT) with F frequency bins, and apply a time-varying filter $\mathbf{y}_t \in \mathbb{R}_+^F$ to the magnitude spectrum $\mathbf{m}_t$ of the mixture to obtain an estimate $\hat{\mathbf{s}}_t$ of the speech magnitude spectrum such that:

$$\hat{\mathbf{s}}_t^\alpha = \mathbf{y}_t \otimes \mathbf{m}_t^\alpha \tag{1}$$

where $\otimes$ denotes element-wise multiplication and $\alpha > 0$ is an exponent that affects the estimation of $\mathbf{y}_t$. A time-domain signal is then reconstructed using inverse STFT of the complex spectrum obtained from $\hat{\mathbf{s}}_t$ and the phase of the mixture.

In many cases, it is useful to estimate filters in a reduced resolution feature space, for example obtained using a Mel transform. An advantage of this is that the filters may be smoother and easier to learn, requiring fewer parameters, and might generalize better to unseen speakers and noise [3], despite reducing the achievable separation quality. See [3] for a comparison of Mel-domain with full-resolution speech enhancement based on NMF.

We consider a Mel transformation applied to the full-resolution spectrum as $\mathbf{m}_t^{\text{mel}} = \mathbf{B}\mathbf{m}_t^\alpha$ with $\mathbf{B} = (b_{i,f}) \in \mathbb{R}^{B \times F}$, where B is the number of Mel bins and $b_{i,f}$ is the weight of the DFT bin f in the i-th Mel bin, and similarly for $\mathbf{s}^{\text{mel}}$ and $\mathbf{n}^{\text{mel}}$. From a filter estimated in that domain, we have to estimate a corresponding full-spectrum filter to use with (1). However, the Mel matrix $\mathbf{B}$ is rectangular ($B < F$) and hence the corresponding linear transform is not invertible. As an 'ad-hoc' method to reconstruct from Mel domain filters, we can compute a full-spectrum filter as:

$$\mathbf{y}_t = \mathbf{B}^\intercal \mathbf{y}_t^{\text{mel}}. \tag{2}$$

Due to the fact that the rows of $\mathbf{B}$ are overlapping Mel filter envelopes that sum to one, this distributes the estimated filter value $\mathbf{y}_{i,t}^{\text{mel}}$ for the i-th Mel filter back to the f-th full-spectrum frequency bin in proportion to that bin's original contribution $b_{i,f}$ to that Mel filter. Although this is a rather ad-hoc approach, we found that it did not perform worse in terms of SNR than a more principled approach using a Wiener-like filter, where the Mel-domain speech and noise estimates are both transformed with the pseudo-inverse $\mathbf{B}^+$ of $\mathbf{B}$.

III. Supervised training for speech separation

The most common approach to estimate the filter $\mathbf{y}_t$ is based on time-frequency masking [1]–[9], which restricts the filter to $[0, 1]^F$ to form a time-frequency mask. This resctriction is reasonable: it introduces little approximation error (0.36 dB in oracle experiments), and avoids estimation of unbounded values. These methods rely on a supervised training scheme based on a parallel training corpus of clean speech signals and speech mixtures. They optimize a system $\mathbf{m}_t \mapsto \hat{\mathbf{y}}_t$ that produces a mask estimate $\hat{\mathbf{y}}_t$ from the features $\mathbf{m}_t$ of the mixed signal. Among these, two main approaches have emerged: the *mask approximation* approach trains the system so that

the estimated mask best approximates a reference mask computed using the clean and noisy speech; the *signal approximation* approach trains the system so that the estimated mask, when applied to the mixture, leads to the best approximation of the reference signal.

In both approaches, it may be useful to introduce a non-linear warping $x \mapsto x^\alpha$ of the magnitudes in the objective function, in order to differentially affect the sharpness of the mask or the dynamic range of the features. Here, we consider $\alpha = 2$ (power spectrum), $\alpha = 1$ (magnitude spectrum) and $\alpha = 2/3$ ('auditory' spectrum). The latter is motivated by the 'power law of hearing' as in computation of perceptual linear prediction (PLP) coefficients [10].

A. Mask approximation (MA)

In mask approximation, given a reference mask $\mathbf{y}_t^*$, the objective function is defined as

$$E^{\mathsf{MA}}(\hat{\mathbf{y}}) = \sum_{f,t} D(\hat{y}_{f,t}, y^*_{f,t}) \tag{3}$$

where D is a distance measure. In this paper, we use the squared Euclidean distance, which ensures that E^{MA} is closely related to the source separation evaluation criterion in terms of signal-to-distortion ratio (SDR). The reference mask is often taken to be the so-called *ideal ratio mask* (IRM) [4]:

$$\mathbf{y}_t^* = \frac{\mathbf{s}_t^\alpha}{\mathbf{s}_t^\alpha + \mathbf{n}_t^\alpha}, \tag{4}$$

where $\mathbf{n}_t$ is obtained from $n(t) = m(t) - s(t)$, and division is performed element-wise.

B. Signal approximation (SA)

Even though the mask approximation objective is discriminative, it does not directly optimize the actual source separation objective, which is to deliver the best possible reconstruction of the speech signal (e.g., in terms of SDR). We use instead the following signal approximation objective, whose minimization maximizes the SNR for the warped features in each time-frequency bin:

$$E^{\mathsf{SA}}(\hat{\mathbf{y}}) = \sum_{f,t} \left(\hat{s}^\alpha_{f,t} - s^\alpha_{f,t}\right)^2 = \sum_{f,t} \left(\hat{y}_{f,t} m^\alpha_{f,t} - s^\alpha_{f,t}\right)^2. \tag{5}$$

Such an objective function can be applied to any mask estimation scheme, for example see [8], [9]. It can in particular be used to estimate a Mel-domain mask $\hat{\mathbf{y}}^{\mathrm{mel}} = (\hat{y}^{\mathrm{mel}}_{i,t})$ by substituting (2):

$$E^{\mathsf{SA,Mel}}(\hat{\mathbf{y}}^{\mathrm{mel}}) = \sum_{f,t} \Big(\big(\sum_i b_{i,f} \hat{y}^{\mathrm{mel}}_{i,t} \big) m^\alpha_{f,t} - s^\alpha_{f,t} \Big)^2, \tag{6}$$

which takes into account the fact that the Mel mask $\hat{y}_{i,t}$ influences one or more DFT bins.

C. Mask estimation by deep neural networks

We now describe the mask estimators considered in this paper. While some studies used Support Vector Machines [5] or decision trees [7], there is an increasing trend towards deep neural network (DNN) based speech separation [4], [6], [9]. In this study, we first use K-layer feed-forward DNNs with $K-1$ hidden layers and one output layer, which compute an estimated mask $\hat{\mathbf{y}}_t$ as

$$\hat{\mathbf{y}}_t = \sigma\left(\mathbf{W}^K \mathcal{H}\left(\mathbf{W}^{K-1} \cdots \mathcal{H}\left(\mathbf{W}^1 [\mathbf{x}_t; 1]\right)\right)\right), \tag{7}$$

where $\mathbf{x}_t$ are the input features, σ denotes the element-wise logistic sigmoid function, $\mathcal{H}$ is an element-wise non-linear function (here we use the hyperbolic tangent), and $[\mathbf{a}; \mathbf{b}] := (\mathbf{a}^\intercal, \mathbf{b}^\intercal)^\intercal$ denotes row-wise concatenation. For our DNN experiments, we concatenate C consecutive frames of log spectra of the mixture ($C-1$ past frames and the current frame, to allow for real-time operation) to obtain the input features $\mathbf{x}_t = \log[\mathbf{m}_{t-C+1}; \cdots; \mathbf{m}_t]$.

Deep neural networks have a few convenient properties for the speech separation task. First, the masking functions for all frequency bins can be represented in a single model. Second, non-linearities in the feature representation can be introduced effectively, thus allowing for compression of the spectral magnitudes, which is considered useful in speech processing. Once trained, (7) can be very efficiently evaluated, unlike iterative methods such as NMF. Finally, the backpropagation algorithm allows for easy discriminative training, since only the gradient of the objective function with respect to the network output $\hat{\mathbf{y}}$ needs to be modified accordingly, whereas all other derivatives are unaffected. In particular, computing the gradients $\partial E^{\mathsf{MA}}/\partial \hat{\mathbf{y}}$, $\partial E^{\mathsf{SA}}/\partial \hat{\mathbf{y}}$ and $\partial E^{\mathsf{SA,Mel}}/\partial \hat{\mathbf{y}}$ is straightforward.

D. Deep recurrent neural networks

Since audio is sequential, it is not surprising that in recent years recurrent neural networks have seen a resurgence in popularity for speech and music processing tasks [9], [11]–[15]. The combination of deep structures with temporal recurrence yields so-called deep recurrent neural networks (DRNNs) [13]. The function computed by deep recurrent neural networks can be defined by the following iteration for $k = 1, \ldots, K-1$ and $t = 1, \ldots, T$:

$$\mathbf{h}_0^{1,\ldots,K-1} = \mathbf{0}, \tag{8}$$

$$\mathbf{h}_t^0 = \mathbf{x}_t, \tag{9}$$

$$\mathbf{h}_t^k = \mathcal{H}(\mathbf{W}^k [\mathbf{h}_t^{k-1}; \mathbf{h}_{t-1}^k; 1]), \tag{10}$$

$$\hat{\mathbf{y}}_t = \sigma(\mathbf{W}^K [\mathbf{h}_t^{K-1}; 1]). \tag{11}$$

In the above, $\mathbf{h}_t^k$ denotes the hidden feature representation of time frame t in the level k units ($k = 0$: input layer (9)).

To train RNNs, the recurrent connections in (10) can be 'unfolded', conceptually yielding a T-layer deep network with tied weights. However, this approach ('backpropagation through time') suffers from a vanishing or exploding gradient for larger T, making the optimization difficult [16]. As a result, RNNs are often not able to outperform DNNs in practical speech processing tasks [9], [17]. One of the oldest, yet still most effective solutions proposed to remedy this problem is to add structure to the RNN following the Long Short-Term Memory (LSTM) principle as defined in [18], [19]. In particular, LSTM-DRNNs perform exceptionally well on standard speech recognition benchmarks [13], [20].

In LSTM networks, the computation of $\mathbf{h}_t^k$ is performed by a differentiable function $\mathcal{L}^k(\mathbf{h}_t^k; \mathbf{h}_{t-1}^k)$ which performs soft versions of read, write, and delete operations on a memory variable. Each of these operations is governed by weights which are optimized in the manner of backpropagation through time. The memory is implemented as a recurrent unit with weight 1, allowing the RNN to preserve an arbitrary amount of temporal context. It can be shown that this approach avoids the vanishing gradient problem, thus allowing to effectively train DRNNs using gradient descent.

E. Baseline: discriminative non-negative matrix factorization

As a strong, model-inspired baseline for supervised speech separation, we use discriminative NMF (DNMF) [8]. At test time, DNMF computes the mask $\hat{\mathbf{y}}_t$ as follows:

$$\mathbf{h}_t^0 = \mathbf{1} \otimes (1/R), \tag{12}$$

$$\mathbf{h}_t^k = \mathbf{h}_t^{k-1} \otimes \frac{\mathbf{W}^\intercal(\mathbf{x}_t / \mathbf{W}\mathbf{h}_t^{k-1})}{\mathbf{W}^\intercal \mathbf{1} + \lambda}, \quad 1 \le k < K, \tag{13}$$

$$\hat{\mathbf{y}}_t = \frac{\sum_{r \le R_s} \mathbf{w}^{K,(r)} h^K_{r,t}}{\mathbf{W}^K \mathbf{h}_t^K} \tag{14}$$

where R is the number of NMF dictionary atoms, $\mathbf{W} = [\mathbf{w}^{(1)} \cdots \mathbf{w}^{(R_s)} \cdots \mathbf{w}^{(R)}]$ and $\mathbf{W}^K = [\mathbf{w}^{K,(1)} \cdots \mathbf{w}^{K,(R_s)} \cdots \mathbf{w}^{K,(R)}] \in \mathbb{R}_+^{CF \times R}$ are NMF dictionaries with R_s speech atoms and $R - R_s$ noise atoms, each of which corresponds to a sliding window of C contiguous STFT spectra (magnitude, $\alpha = 1$). $\mathbf{x}_t \in \mathbb{R}_+^{CF}$ is a sliding window of mixture magnitude spectra similar to the input features of the DNN, λ is a free parameter controlling the sparsity of the 'hidden' activations $\mathbf{h}$, and K is a fixed number of iterations.

In conventional NMF, it is assumed that $\mathbf{W}^K = \mathbf{W}$, and $\mathbf{W}$ is trained non-discriminatively, for example using sparse NMF on each source [21]. Note that, as shown in [8], sparse NMF can significantly outperform the recently popular 'exemplar-based' approaches [3] based on random sampling of speech and noise observations.

However in the context of discriminative training, it is convenient and effective to allow $\mathbf{W}^K$ to differ from $\mathbf{W}$, so that $\mathbf{W}^K$ can be trained using the objective function (5), given the activations $\mathbf{h}_t^K$ obtained by (13). A multiplicative update algorithm for this optimization is given in [8].

IV. Experimental Setup

Our methods are evaluated on the corpus of the 2nd CHiME Speech Separation and Recognition Challenge (track 2: medium vocabulary) [22], which is publicly available[1]. The task is to estimate speech embedded in noisy and reverberant mixtures. Training, development, and test sets of noisy mixtures along with noise-free reference signals are created from the Wall Street Journal (WSJ-0) corpus of read speech and a corpus of noise recordings. The noise was recorded in a home environment with mostly non-stationary noise sources such as children, household appliances, television, radio, etc. The dry speech recordings are convolved with a time-varying sequence of room impulse responses from the same environment where the noise corpus is recorded. The training set consists of 7 138 utterances at six SNRs from -6 to 9 dB, in steps of 3 dB. The development and test sets consist of 410 and 330 utterances at each of these SNRs, for a total of 2 460 and 1 980 utterances. Our evaluation measure for speech separation is source-to-distortion ratio (SDR) [23]. By construction of the WSJ-0 corpus, our evaluation is speaker-independent. Furthermore, the background noise in the development and test set is disjoint from the noise in the training set, and a different room impulse response is used to convolve the dry utterances.

All experiments use spectral features obtained with the square root of the Hann window, a frame size of 400 samples (25 ms) and a frame shift of 160 samples (10 ms). For the NMF baseline, we set $C = 9$, $K = 25$, $R_s = 1\,000$, $R = 2\,000$ and $\lambda = 5$ based on limited parameter tuning on the CHiME development set [8].

In D(R)NN training, all the weight matrices $\mathbf{W}^k, k = 1, \ldots, K$ are estimated by supervised training as outlined in Section III. The training targets are derived from the parallel noise-free and multi-condition training sets of the CHiME data. The input features are globally mean and variance normalized on the training set, this kind of normalization allowing for on-line processing at run time. The DNN topology was optimized based on limited parameter tuning (number of hidden layers and units) on the CHiME development set (cf. Table I). The DRNN topology used in this study was determined based on earlier experiments with speech separation and feature enhancement on different corpora. All weights are randomly initialized with Gaussian random numbers ($\mu = 0$, $\sigma = 0.1$). For DNN training, 'discriminative' pre-training is used [24], i.e., building the DNN layer by layer by backpropagation (as opposed to generative pre-training).

TABLE I
Average SDR for various topologies (# of hidden layers × # of hidden units per layer) of DNN and LSTM-DRNN on the CHiME development set.

SDR [dB]	Input SNR [dB]						
	-6	-3	0	3	6	9	Avg.
Noisy	-3.73	-1.05	1.18	2.86	4.53	6.19	1.66
DNN 1×1024	4.48	6.90	8.96	10.38	12.11	13.95	9.46
DNN 2×1024	4.76	7.17	9.15	10.62	12.38	14.27	9.72
DNN 3×1024	5.77	8.00	9.92	11.24	12.99	14.84	**10.46**
DNN 4×1024	5.70	7.92	9.91	11.26	13.02	14.83	10.44
DNN 2×1536	4.61	7.06	9.13	10.60	12.39	14.28	9.68
LSTM-DRNN 1×256	7.30	9.31	11.14	12.38	14.15	15.93	11.70
LSTM-DRNN 2×256	7.94	9.89	11.68	12.92	14.60	16.35	**12.23**
LSTM-DRNN 3×256	7.64	9.69	11.52	12.70	14.46	16.18	12.03
Oracle (IRM)	*13.91*	*15.26*	*16.52*	*17.38*	*18.91*	*20.49*	*17.08*

We train the DNNs and DRNNs through stochastic ('on-line') gradient descent with an initial learning rate of 10^{-5} and a momentum of 0.9. Weights are updated after 'mini-batches' of 25 feature sequences. In DRNN training, sequences within these mini-batches are processed in parallel on a graphics processing unit (GPU), but unlike in DNN training, there is no parallelism across time steps. Hence, to increase the efficiency of DRNN training, the utterances are 'chopped' into sequences of at most $T = 100$ timesteps (but not shorter than $T = 50$).

Two common strategies are used to reduce over-fitting on the training set. First, Gaussian noise ($\mu = 0$, $\sigma = 0.1$) is added to the inputs in the training phase. Second, we use an early stopping strategy where we evaluate the objective function on the development set after each training epoch and select the best network accordingly. Training is stopped as soon as no improvement on the development set is observed for ten training epochs or after 100 epochs. We use the GPU enabled DNN and LSTM-DRNN training software CURRENNT [25], which is publicly available[2].

V. Results and Discussion

A. Neural network topologies

Table I shows the source separation performance using various network architectures and dimensions. Best DNN results are obtained with 3 layers and 1024 units per layer (10.46 dB SDR), whereas for 4 layers the performance saturates. 1.0 dB SDR is gained by increasing the depth from 1 to 3 layers, whereas increasing the width of the network to 1536 units does not seem to help. LSTM-DRNN can achieve up to 12.23 dB SDR with a much smaller model size (3×1024 DNN: 4.1 M trainable parameters, 2×256 LSTM-DRNN: 1.0 M), indicating a clear benefit of explicitly modeling temporal dependencies. Interestingly, the benefit of adding depth to LSTM-DRNN (besides their inherent depth in time) seems to be comparatively minor for the de-noising task, leading to competitive results even with a single layer (11.70 dB).

B. Influence of feature representation

Fig. 1 shows the influence of the feature representation on the oracle masking performance as well as on the results obtained with supervised training of mask estimation with LSTM-DRNNs. As is expected, in the oracle case the full-resolution mask delivers the best SDR. Regarding warping, $\alpha = 1$ (magnitude spectrum) works best. However, when the estimated mask is used, best results are

[1] http://spandh.dcs.shef.ac.uk/chime_challenge/ – as of July 2014

[2] https://sourceforge.net/p/currennt

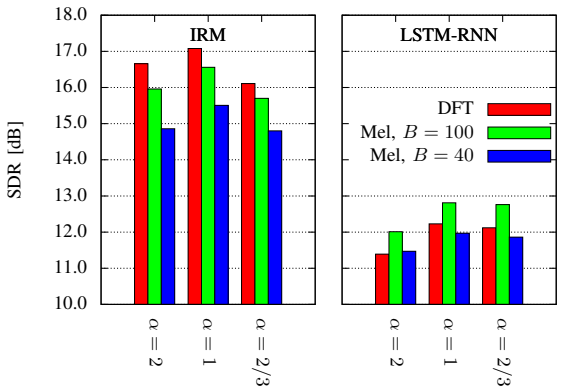

Fig. 1. SDR on CHiME development set with oracle masking (ideal ratio mask, IRM) as well as LSTM-DRNN based mask approximation (MA) for various values of the spectral warping parameter α used in computation of DFT and Mel spectra ($B = 40$, $B = 100$).

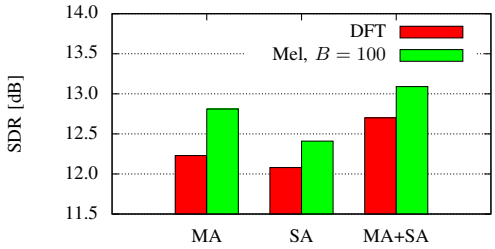

Fig. 2. SDR on the CHiME development set with LSTM-DRNN mask estimation, trained with the mask approximation (MA) and signal approximation (SA) objectives, and SA-based retraining of LSTM-DRNNs trained with MA (MA+SA). Mel ($B = 100$) and DFT magnitudes ($\alpha = 1$).

obtained with Mel masks ($B = 100$), and the full-resolution mask works only slightly better than the low-resolution ($B = 40$) Mel mask. Since for $B = 100$, the lower Mel bins correspond to single DFT bins while the higher Mel bins comprise multiple DFT bins, this indicates difficulties in precisely estimating the mask for the higher frequencies, which could be due to insufficient training data. Furthermore, while 'auditory' spectra ($\alpha = 2/3$) deliver clearly the worst performance in oracle masking, they are on par with magnitude spectra for the estimated mask. Apparently, using warping with $\alpha = 2/3$ (which smoothes the training targets) eases the optimization of the cost function enough to compensate for the lower attainable performance in oracle masking. Overall, the performance differences stemming from the feature representation are surprising. In the DFT power spectrum domain, 11.39 dB average SDR are obtained while in the Mel magnitude domain ($B = 100$) we get 12.81 dB.

C. Influence of the objective function

Fig. 2 shows the impact of using discriminative objective functions for $\alpha = 1$. Interestingly, when training LSTM-DRNNs using the discriminative objectives E^{SA} and $E^{\mathsf{SA,Mel}}$ ('SA' in Fig. 2), we obtain worse performance than with mask approximation ('MA' in Fig. 2). We found sub-optimal convergence of the cost function in this case, both on the training and held-out development set. However, if we start from the solution obtained by training with E^{MA} until convergence, we can significantly improve the results over MA ('MA + SA' in Fig. 2). Yet the results in the DFT domain using MA + SA are still below the results with Mel domain MA. Furthermore, if we apply MA + SA in the Mel domain, we can obtain best results (13.09 dB average SDR on the CHiME development set).

TABLE II
SOURCE SEPARATION PERFORMANCE FOR SELECTED SYSTEMS ON CHiME TEST SET ($\alpha = 1$). Mel: $B = 100$.

SDR [dB]	Mel	SA	Input SNR [dB]						
			-6	-3	0	3	6	9	Avg.
Noisy			-2.27	-0.58	1.66	3.40	5.20	6.60	2.34
NMF [8]			5.48	7.53	9.19	10.88	12.89	14.61	10.10
DNMF [8]		✓	6.61	8.40	9.97	11.47	13.51	15.17	10.86
DNN			6.89	8.82	10.53	12.25	14.13	15.98	11.43
DNN		✓	7.89	9.64	11.25	12.84	14.74	16.61	12.16
DNN	✓	✓	8.36	10.00	11.65	13.17	15.02	16.83	12.50
LSTM-DRNN	✓	✓	10.14	11.60	13.15	14.48	16.19	17.90	**13.91**
Oracle (IRM)		–	*14.53*	*15.64*	*16.95*	*18.09*	*19.65*	*21.24*	*17.68*
Oracle (IRM)	✓	–	*14.00*	*15.14*	*16.45*	*17.62*	*19.21*	*20.82*	*17.21*

D. CHiME test set evaluation

We conclude our evaluation with a comparison of selected speech enhancement systems on the CHiME test set, cf. Table II. The topologies for DNN and LSTM-DRNNs as tuned on the development set are used (2×256 LSTM-DRNN and 3×1024 DNN, cf. Table I). The default training procedure for DNN is MA, while the training procedure for DNN and LSTM-DRNNs with SA is MA+SA as described above. Comparing the results obtained with full-resolution magnitude spectra, we observe that considering signal approximation in the objective leads to a performance improvement for both DNN and NMF. Note that DNN including SA-based training outperformed the DNMF results reported in [8], but it remains to be seen how the methods would compare with similar training procedures, e.g., MA+SA, use of the Mel domain, and optimization of α. As on the development data, using the Mel magnitude domain ($B = 100$) instead of DFT improves the results for the DNN. The gains by using the LSTM-DRNN network architecture are complementary, and 1.4 dB performance improvement are achieved with the LSTM-DRNN over a strong DNN baseline using Mel magnitudes and SA-based discriminative training, leading to the best result of 13.91 dB average SDR. While this corresponds to 11.6 dB gain over the noisy baseline, there is still a gap of 3.77 dB relative to the oracle masking (17.68 dB). Audio examples are available at http://www.mmk.ei.tum.de/%7Ewen/denoising/chime.html.

VI. CONCLUSIONS

By a comparative evaluation on the CHiME Challenge data set, we were able to show that a straightforward discriminative training criterion based on optimal speech reconstruction can improve the performance of time-frequency masking approaches to speech separation. Best performance in real-time speech separation on the CHiME database was achieved by discriminatively trained DRNNs operating in the Mel domain. It is interesting that DRNNs outperform DNNs by a large margin in our study, whereas this was not the case in earlier work [9]; we attribute this to avoiding the vanishing temporal gradient in conventional DRNN training as used by [9] thanks to the LSTM architecture. Furthermore, it is notable that the choice of feature representation has such a strong effect on the results, but this is in accordance with earlier studies showing that DNN acoustic models cannot compensate even for simple rotations of the input features [26]. In future work, we will investigate whether the lack of training data may have been responsible for the under-performance of full-resolution features. Such features could indeed support the separation of harmonics in the higher frequencies.

REFERENCES

[1] B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh, "Non-negative matrix factorization based compensation of music for automatic speech recognition," in *Proc. of INTERSPEECH*, Makuhari, Japan, 2010, pp. 717–720.

[2] F. Weninger, J. Feliu, and B. Schuller, "Supervised and semi-supervised supression of background music in monaural speech recordings," in *Proc. of ICASSP*, Kyoto, Japan, 2012, pp. 61–64.

[3] D. Baby, T. Virtanen, T. Barker, and H. Van hamme, "Coupled dictionary training for exemplar-based speech enhancement," in *Proc. of ICASSP*, Florence, Italy, 2014, pp. 2907–2911.

[4] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. of ICASSP*, Vancouver, Canada, 2013, pp. 7092–7096.

[5] J. Le Roux, S. Watanabe, and J. Hershey, "Ensemble learning for speech enhancement," in *Proc. of WASPAA*, Oct. 2013.

[6] F. Weninger, F. Eyben, and B. Schuller, "Single-channel speech separation with memory-enhanced recurrent neural networks," in *Proc. of ICASSP*, Florence, Italy, 2014, pp. 3737–3741.

[7] S. Gonzalez and M. Brookes, "Mask-based enhancement for very low quality speech," in *Proc. of ICASSP*, Florence, Italy, 2014, pp. 7079–7083.

[8] F. Weninger, J. Le Roux, J. Hershey, and S. Watanabe, "Discriminative NMF and its application to single-channel source separation," in *Proc. of INTERSPEECH*, Singapore, Singapore, 2014, to appear.

[9] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Proc. of ICASSP*, Florence, Italy, 2014, pp. 1581–1585.

[10] H. Hermansky, "Perceptual linear predictive analysis for speech," *The Journal of The Acoustical Society of America (JASA)*, vol. 87, pp. 1738–1752, 1990.

[11] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *Proc. of ICASSP*, Kyoto, Japan, 2012, pp. 121–124.

[12] A. Maas, Q. Le, T. O'Neil, O. Vinyals, P. Nguyen, and A. Ng, "Recurrent neural networks for noise reduction in robust asr," in *Proc. of INTERSPEECH*, Portland, OR, USA, 2012.

[13] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. of ICASSP*, Vancouver, Canada, May 2013, pp. 6645–6649.

[14] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proc. of ICML*, J. Langford and J. Pineau, Eds., Edinburgh, Scotland, 2012, pp. 1159–1166.

[15] C. Weng, D. Yu, S. Watanabe, and B.-H. Juang, "Recurrent deep neural networks for robust speech recognition," in *Proc. of ICASSP*, Florence, Italy, 2014, pp. 5569–5573.

[16] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[17] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll, "Feature enhancement by deep LSTM networks for ASR in reverberant multisource environments," *Computer Speech and Language*, vol. 28, no. 4, pp. 888–902, 2014.

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

[19] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[20] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. of ICML*, Beijing, China, 2014.

[21] P. D. O'Grady and B. A. Pearlmutter, "Discovering convolutive speech phones using sparseness and non-negativity," in *Proc. of ICA*, ser. Lecture Notes in Computer Science, M. E. Davies, C. J. James, S. A. Abdallah, and M. D. Plumbley, Eds. Springer Berlin Heidelberg, 2007, vol. 4666, pp. 520–527.

[22] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME' speech separation and recognition challenge: Datasets, tasks and baselines," in *Proc. of ICASSP*, Vancouver, Canada, 2013, pp. 126–130.

[23] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

[24] D. Yu, L. Deng, F. Seide, and G. Li, "Discriminative pretraining of deep neural networks," US Patent 13/304 643, 2011, pending.

[25] F. Weninger, J. Bergmann, and B. Schuller, "Introducing CURRENNT – the Munich open-source CUDA RecurREnt Neural Network Toolkit," *Journal of Machine Learning Research*, vol. 15, 2014, in press.

[26] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proc. of ICASSP*, Kyoto, Japan, 2012, pp. 4273–4276.