



Deep Speech Denoising with Vector Space Projections

Jeffrey Hetherly, Paul Gamble, Maria Barrios, Cory Stephenson, Karl Ni

Lab41, In-Q-Tel

{jhetherly, pgamble, mbarrios, cstephenson, kni}@iq.t.org

Abstract

We propose an algorithm to denoise speakers from a single microphone in the presence of non-stationary and dynamic noise. Our approach is inspired by the recent success of neural network models separating speakers from other speakers and singers from instrumental accompaniment. Unlike prior art, we leverage embedding spaces produced with source-contrastive estimation, a technique derived from negative sampling techniques in natural language processing, while simultaneously obtaining a continuous inference mask. Our embedding space directly optimizes for the discrimination of speaker and noise by jointly modeling their characteristics. This space is generalizable in that it is not speaker or noise specific and is capable of denoising speech even if the model has not seen the speaker in the training set. Parameters are trained with dual objectives: one that promotes a selective bandpass filter that eliminates noise at time-frequency positions that exceed signal power, and another that proportionally splits time-frequency content between signal and noise. We compare to state of the art algorithms as well as traditional sparse non-negative matrix factorization solutions. The resulting algorithm avoids severe computational burden by providing a more intuitive and easily optimized approach, while achieving competitive accuracy.

Index Terms: deep learning, speech, speaker denoising, non-stationary processes

1. Introduction

Signal denoising has been a problem in multiple media for over a century with applications ranging from acoustic speech processing, image processing, seismic data analysis, and other modalities. For each application, approaches have evolved over the span of several decades ranging from traditional statistical signal processing like Wiener and Kalman filtering, wavelet theory, and specific instances of matrix factorization. While effective for locally as well as wide-sense stationary signals, even with a storied history, these efforts have seen less success with more dynamic and *in-the-wild* sets of noise owing to their algorithmic capacity.

Dynamic noise represents many real-world speech situations, and solutions that have impressive results primarily focus on hardware: array processing efforts [1] in the form of SONAR, RADAR, and Synthetic Aperture sensing. These methods solve the problem by using the inputs of multiple sensors to process the source of interest. Unfortunately, much of recorded contemporary media is typically done with more limited hardware, such as a smartphone, and the same methods cannot be easily extended to the monaural case [2], in which audio is recorded from a single microphone.

In previous approaches to the monaural problem, assumptions are explicitly made on signal and noise attributes, or specifications have enabled some control over environment and recording devices. The more general case of single-track speech

recordings in noisy or reverberant rooms has become increasingly common due to the proliferation of inexpensive portable devices with recording capabilities such as cellphones and laptops. In such cases, no guarantees on speech or noise attributions can be assumed regarding the nature of the environment or the location of microphones.

Over the last decade, machine learning approaches have begun to see success in this scenario. In particular, adaptation of familiar matrix factorization techniques [3] to the processing of time-frequency representations of audio signals has proven useful. However, these methods can be difficult to make performant [4], and in many cases additional complexity is required to model source characteristics accurately.

More complex sources can be modeled with the inclusion of *a priori* knowledge regarding their characteristics. This can be empirically derived from a large corpus of training data provided the model in use has a high capacity. In recent years, neural network and deep learning approaches have had great success in other audio processing applications, including the general speech denoising problem [5]. Among these methods, recurrent neural networks in particular have shown the most promise in modeling acoustic time series [6, 7], especially when applied to time-dependent spectral features.

One challenging aspect of neural network approaches is the development of cost functions. For speech signals in particular, the computational complexity of the cost function is important as the timescales associated with speech contain many samples. Additionally, if the goal is to separate sources (e.g. a speaker and a noise source), the cost function must be invariant to different permutations of the recovered sources since the ordering is arbitrary. The proposed approach automates the featurization of speakers and the characterization of noise using an efficient permutation invariant sampling technique.

Building upon our previous source-contrastive estimation work [8], we extend our approach of directly optimizing a vector space that isolates specific source characteristics to the general speech denoising problem. Additionally, we further improve upon our model by incorporating a mask inference term in the cost function as proposed in [7]. Our model is capable of removing a wide variety of noise types from speech and is computationally efficient to train.

The remainder of this paper describes our approach to finding optimal vector spaces using SCE and applies the technique to synthetically-mixed noisy speech. We dive deeper into the state of the art techniques, some of which we leverage, in Sec. 2. The approach is then described in Sec. 3 with implementation details in Sec. 4. Experimental results are shown in Sec. 5, which is followed by a summary and discussion of future work.

2. Related Work

Considerable work has been done in speech processing for as long as recordings have been made. Here we describe a few approaches representative of the diverse set of solutions to the

denoising problem, though many others (such as [9]) exist and have been omitted for brevity. We begin with methods that are rooted in signal processing theory, where a large number of approaches use some type of matrix factorization [10, 11]. In particular, sparse non-negative matrix factorization (SNMF) is shown effective at extracting non-stationary noise sources in [4, 12, 3]. SNMF constructs a set of spectral basis functions from training data and linearly combines these with a set of learned weights to reconstruct the spectral features of the desired signal. Sparsity is typically enforced by an ℓ_1 -norm constraint on the learned weights that contains a multiplicative hyper-parameter, μ . As will be shown in Sec 5, linear methods such as these lack the algorithmic capacity to compete with more modern techniques.

2.1. Convolutional Denoising Autoencoder

Autoencoders have been used to successfully remove noise and to isolate single sources from audio signals [13]. At a high level, autoencoders learn to featurize inputs (encoding) and then reconstruct them as outputs (decoding). This approach is well suited to denoising because a model is forced to build compressed representations of the input by discarding information about the noise.

Closely related are convolutional autoencoders, typically used for denoising images [14, 15]. These models make use of convolutional layers during encoding and deconvolutional layers during decoding. Convolutional denoising autoencoders (DAE) applied to audio signals represented via spectrograms (using STFT) operate similarly, though many of these approaches have problems generalizing to unseen signals. Moreover, convolutional autoencoders are an architectural construct still dependent on their cost functions, a challenge that defines how they perform in the context of denoising, where the common ℓ_2 -norm may not be sufficiently descriptive.

2.2. Neural Network-Based Source Embeddings

Recent success in monaural audio source separation and denoising have taken to learned embedding vectors [6, 16, 17, 7]. The primary advantage of learning embedding vectors is that they bypass the so-called permutation problem in which the output of a learning algorithm must be permuted to account for the unordered nature of the target sources [18]. Additionally, the number of sources to be separated and denoised can be arbitrary with an appropriate clustering technique (although, this depends on how inference is performed).

The embedding model we propose in the following sections most resembles deep clustering [6] and mask inference (DC+MI) found in [7], though with a vastly reduced cost function. The DC+MI network learns embeddings given the spectral magnitude of the mixed audio sample using a series of four bi-directional LSTMs (BLSTM). In addition to clustering those embeddings to create a binary mask as in [6], a learned non-linear transformation is used to directly translate the embeddings into a ratio mask. This has the advantage of limiting some of the artifacts inherent to performing a binary mask. However, this comes at a cost of fixing the number of sources to two. Clustering on an arbitrary number of sources can still be performed on the embeddings, but only a binary mask can be constructed from these clusters.

3. Approach

The approach used in this paper combines the mask inference capabilities of cited literature in Sec. 2 with the flexibility of SCE to remove dynamic, non-stationary noise sources from speech for monaural audio signals.

3.1. Datasets

Our task is to isolate speech from a mixture of dynamic noise and speech using a monaural audio signal. All denoising algorithms are trained and evaluated on a mixture of the LibriSpeech [19] and UrbanSound8K [20] datasets. LibriSpeech provides high-quality audio recordings of isolated English speech from both male and female speakers and UrbanSound8K provides recordings from ten non-stationary noise classes. Two two-second clips from each dataset are added at various SNR ratios to create the noisy-speech data. The SNR ratio is continuously varied between -5 and 5 dB for the training phase for all but SNMF algorithm wherein speech and noise are fed in separately. No impulse response convolution is used so as to focus solely on removing non-stationary sources of noise. The application of our technique to speech in reverberant environments is left for future work.

For the training, validating, and in-set testing of each algorithm we use the train-clean-100 set of audio readings from the LibriSpeech dataset, which provides approximately 100 hours of speech evenly split between female and male speakers. For out-of-set testing we use the dev-clean set from LibriSpeech. Although all noise types from UrbanSound8K are used for training, noise files from each noise type are reserved for training, validation, and testing.

3.2. Model

Our model for denoising monaural signals assumes that additive noise is linearly mixed with speech signal and can be separated into individual source components. In this context, a source is either a speaker or a particular type of noise. For a given source i in a speaker-noise mix, our model masks the magnitude response. This mask filters out information from time-frequency bins in the short-time Fourier transform (STFT), $X(t, f)$, that do not belong to a given source, while passing those time-frequency bins that do.

Typically, the predicted mask $Y_{t,f}^{(i)}$ for the i^{th} source is implemented as either a ratio or in our case, a binary mask. We let $Y_i \in \{-1, 1\}^{M \times TF}$, where $M \leq C$, C being the total number of sources in our training set and M being the number to be mixed. To set our masks, if i^{th} source is the loudest in time frequency bin (t, f) , then $Y_{t,f}^{(i)} = 1$, and $Y_{t,f}^{(i)} = -1$ otherwise.

Similar to our previous work [8] and inspired by natural language processing efforts [21], we use a speaker embedding, optimized by the use of two vector spaces: an input and an output space. The first vector space is an input embedding that *implicitly* defines a source, and it is not associated with any source in particular. We also have an output embedding that *explicitly* trains to a corpus of known sources. Inference uses only the input vector space to generalize to any possible source by clustering our neural network outputs. In our notation, the input and output vector spaces for a given sample are implemented as tensors with an embedding space of E , labeled as $V_i(t, f)$ and V_o , respectively. The columns of either tensor have E dimensions (hidden units) and denote the vectors associated with a given source.

To train and generate our embeddings, we use a recurrent

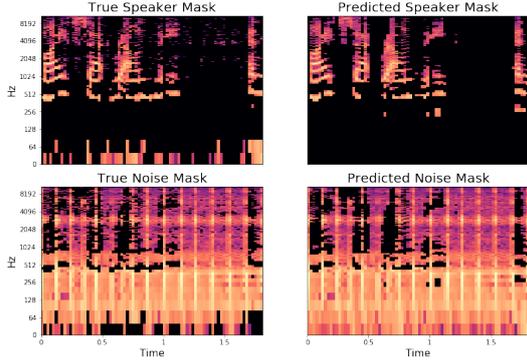


Figure 1: Example of predicted and true binary masks applied to a sample mixed at an SNR of -1

neural network regression to V_i . To compare to [6] and [7], we use a total of four BLSTM layers, and we have a dense layer that is convolved over the output 2D vector produced by the final BLSTM. This final layer of source embeddings is also fed through a non-linear transform as in [7] to yield the ratio mask.

The SCE loss [8] for a given time frequency bin and sample b is denoted as $\mathcal{L}_{t,f}^{(b)}$. Then,

$$\mathcal{L}_{t,f}^{(b)}(\mathbf{v}_i, \mathbf{v}_o) = \frac{-1}{M} \sum_{s \in S_b} \log \sigma \left(Y_{t,f}^{(s,b)} \cdot \mathbf{v}_i^{(b)}(t, f)^T \mathbf{v}_o^{(s)} \right) \quad (1)$$

Here, S_b is the set of sources sampled for the input mix b , and s is a single source from the subset. The total loss is the average of $\mathcal{L}_{t,f}^{(b)}$ over all time frequency bins and all B inputs in the batch.

$$\mathcal{L}(\mathbf{v}_i, \mathbf{v}_o) = \frac{1}{B} \sum_b \sum_{(t,f)} \mathcal{L}_{t,f}^{(b)} \quad (2)$$

Intuitively, the output of the neural network at time (t, f) is $\mathbf{v}_i(t, f)$ and the output vector $\mathbf{v}_o^{(s)}$ is an embedding for source s at frequency f . Say that source 1 is louder than source 2 at time frequency bin (t, f) for sample b . Then we would ideally like the correlation between the embedding produced by our neural network \mathbf{v}_i and the vector for source 1 to be high. That is to say, we would like $\sigma(\mathbf{v}_i^T \mathbf{v}_o^{(1)}) \rightarrow 1$. Simultaneously, the correlation between \mathbf{v}_i and the vector for source 2 should be low, since these two vectors should be anti-correlated if they are sufficiently different. That is to say, we would like $\sigma(\mathbf{v}_i^T \mathbf{v}_o^{(2)}) \rightarrow 0$. Mathematically speaking, we are pulling our embedding towards our source vector $\mathbf{v}_o^{(1)}$ and pushing it away from non-source vectors $\mathbf{v}_o^{(2)}$. Which sources to attribute appropriate correlation/anti-correlation to is determined by the label Y , which will be +1 in the former case and -1 in the latter. It is important to note that we can save on both computation and accuracy by optimizing only those sources that are in S_b , which in our case will have two elements (one speaker and one noise source).

Additionally, during inference we do not use the output vector space V_o . While it is true that computations are further reduced, the intention is that the out-of-set sources set is allowed. In fact, even though we may train on mixes with fewer sources, we can do inference in situations where there are arbitrary numbers of sources.

Our algorithm (denoted SCE+MI) is implemented in Tensorflow, v1.4 [22], with an architecture consisting of four BLSTM layers r_1, r_2, r_3, r_4 of 500 units each. These are followed by a fully connected layer d_1 that maps the output of the fourth BLSTM layer to the input vector space. The BLSTM layers use tanh nonlinearities, and the fully connected layer is linear. For a batch of inputs \mathbf{X} , the output of the four BLSTM layers $r_1, r_2, r_3, r_4 \in \mathbb{R}^{B \times T \times 500}$. While the final (embedding) layer of the neural network is technically a fully-connected linear layer, it is implemented as a 1D convolution over the r_4 output tensor with a filter $w \in \mathbb{R}^{1 \times 500 \times F \cdot E}$. The output of the convolution can then be reshaped to give the input vector space $\mathbf{V}_i \in \mathbb{R}^{B \times T \times F \times E}$. The vector-space output is fed through what is effectively a 1D convolution along the embedding dimension with a softmax that yields the final ratio mask output. This implementation allows the model to be run for arbitrary input T , which is useful at inference time.

For efficient evaluation of the cost function of Eq. 1 across batches, the sources vectors for sources only represented in each batch are assembled into a tensor $\mathbf{V}_o \in \mathbb{R}^{B \times M \times E}$. The ordering of the M speakers in \mathbf{V}_o must match the ordering used in \mathbf{Y} , but is otherwise arbitrary. To efficiently compute the dot products $V_i \cdot V_o$ in Eq. 1 with broadcasting, we expand the dimensions appropriately.

This gives an output of the dot product operation as a tensor $\mathbf{D} \in \mathbb{R}^{B \times T \times F \times M}$, which is compatible with the labels \mathbf{Y} and so they can be multiplied together elementwise to give the argument of the sigmoid in Eq. 1. The remaining portion of the cost function is easily evaluated.

Our batch size is $B = 256$ during training. The input tensors have dimensions $\mathbf{X} \in \mathbb{R}^{B \times T \times F}$ and label tensors are $\mathbf{Y} \in \mathbb{R}^{B \times T \times F \times M}$, where $T = 78$ is the length of total time steps per sample and $F = 257$ are the number of frequency bins used.

4. Experiments

In all experiments, signals are resampled and scaled to 10kHz, zero mean and unit standard deviation, from which the short time Fourier transform (STFT) spectrograms are extracted with a Hanning window of 512 and 256 stride length. We use audio clips of approximately two seconds which, when combined with the STFT operation, yield input features of dimension 257×78 (frequencies by time frames). The complex phases $\phi_{t,f}$ were saved separately for use in post separation processing. Separate spectrograms for the signal from each speaker and noise ($S_{t,f}^{(n)}$ for $n \in \{1, 2, \dots, C\}$) were computed for training and evaluation purposes, while the total spectrogram was computed by the elementwise sum $X_{t,f} = S_{t,f}^{(n)} + S_{t,f}^{(m)}$ for a speaker and noise with IDs n, m .

The magnitudes of the $X_{t,f}$ spectrograms were then passed through a square root nonlinearity and percent normalized. This is similar to the procedure suggested in [23]; however we obtained better results with a square root rather than a logarithmic nonlinearity. Source labels $Y_{t,f}^{(c)}$ are assigned to each T-F bin by giving a value of 1 to the signal with the largest magnitude at that time and frequency, and a value of -1 to all other sources.

4.1. Algorithm Comparisons

We compare three approaches against the proposed work: a linear matrix factorization method (SNMF), a denoising auto-encoder (DAE), and a hybrid deep clustering/mask inference architecture (DC+MI). SNMF is adopted from [4] with the most

optimal hyperparameter settings found therein and trained on 10000 two-second audio clips of noise and speakers. To aid in training SNMF we removed portions of each spectrogram based on a log max-amplitude threshold for each time frame. This threshold was found to isolate spoken words while trimming the surrounding empty audio.

Our comparison convolutional DAE is based on [24] and consists of 15 convolutional layers, followed by 15 deconvolutional layers. Each layer contains 128 5x5 filters with relu activation and constant input size. Skip connections are employed between every other pair of matched convolutional and deconvolutional layers. The model was trained using RMSprop with Nesterov momentum and a learning rate of 5e-5.

The DC+MI network is an implementation of the architecture found in [7]. We use the same optimal set of hyperparameters as they do except that our loss function for the mask inference head uses the true spectral component rather than a proxy.

Comparisons of the performance of each algorithm are quantified by improvement in the source-to-distortion ratio (SDR). Each algorithm is evaluated on how well it improves the SDR metric for an input SNR range of [-5, 5]dB and for each noise type.

4.2. Reconstruction

At inference time for our model and the deep-clustering head of DC+MI, a signal consisting of an unknown mixture of sources is preprocessed as described in the previous subsection, giving a complex T-F estimate of a single source signal, $\hat{S}_{t,f}$. An input feature is generated and fed through the model to obtain the vectors \mathbf{V}_i . A K -means clustering is then performed on the vectors in order to generate a labeling prediction $\hat{\mathbf{Y}} \in \mathbb{R}^{T \times F \times K}$ in which each T-F element is associated with a cluster label. Here the element $\hat{Y}_{t,f}^{(k)} = 1$ if the associated vector $V_{t,f}$ belongs to the k^{th} cluster, and $\hat{Y}_{t,f}^{(k)} = -1$. While K -means clustering was used for both DC and SCE to allow for a more direct comparison of the two objectives, spherical K -means is more natural for SCE. The labels obtained via clustering can then be used as masks to reconstruct a source $S_{t,f}^{(k)}$ from each of the K clusters. T-F representations of the inferred sources are calculated as the element-wise multiplication of the input spectrogram with the inferred labeling.

$$\hat{S}_{t,f}^{(k)} = X_{t,f} \odot \frac{1}{2} (\hat{Y}_{t,f}^{(k)} + 1) \quad (3)$$

The source spectrogram $\hat{S}_{t,f}^{(k)}$ is then converted (using the inverse STFT) into a source waveform, completing the inference process.

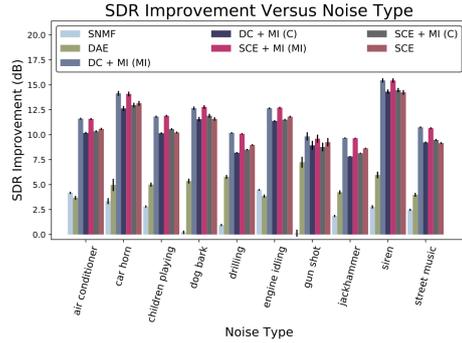
The output of the mask inference head of SCE+MI and DC+MI and that of SNMF is a ratio mask that, when multiplied element-wise with the original spectrogram, yield the respective speaker and noise sources. These ratio masking techniques have the potential to produce higher-quality audio than binary masking as the T-F bins can be shared amongst sources (as is actually the case).

Our contribution, replicated research [7, 4], and evaluation code are open source and can be found at <http://github.com/lab41/magnolia>.

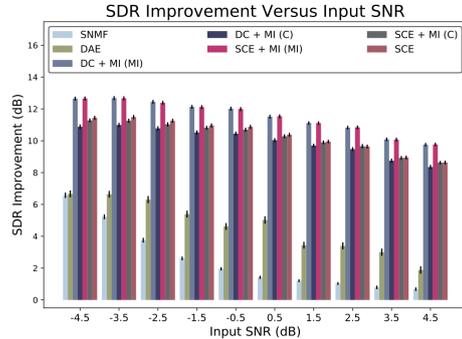
5. Results

The results from our experiments on a hold-out set of mixes are summarized in Figs. 2a and 2b. The performance of the mask

inference head of SCE+MI is on par with the DC+MI (+13 dB at an input SNR of [-5,-4] dB) while the clustering performance of SCE (+11.5 dB) is slightly better than the clustering of the SCE+MI and DC+MI algorithms. Thus, SCE may be more desirable when the number of sources to be separated is arbitrary. The improvements in SDR are greatest for more statistically stationary noise sources and inputs with lower in SNR. This can be explained by the fact that at higher input SNRs the signal is already quite prominent, so there is less room for improvement. The performance of the deep learning-based methods is relatively consistent across input SNRs while SNMF sees more dramatic differences.



(a) Performance versus noise source. DC+MI (C) represents using the embeddings for clustering to reconstruct a binary mask. DC+MI (MI) represents using mask inference for source separation. Likewise for SCE+MI (C) and (MI).



(b) Performance versus input signal-to-noise ratio (SNR). The values indicated for the input SNR represent a range of SNRs ± 0.5 dB around the shown value. (i.e. 3.5dB represents SNRs in the range of [3, 4] dB)

6. Conclusions

We show that SCE with mask inference gives improved reconstruction performance for dynamic noise source denoising. Mask inference performs well (on average, +12 dB in SDR) regardless of the clustering loss it's coupled with. SCE showed the best clustering performance (on average, +11 dB in SDR). This indicates that denoising in the presence of an arbitrary number of sources, SCE may give better accuracy.

7. References

- [1] Josue Sanz-Robinson, Liechao Huang, Tiffany Moy, Warren Rieutort-Louis, Yingzhe Hu, Sigurd Wagner, James C. Sturm, and Naveen Verma, "Robust blind source separation in a reverberant room based on beamforming with a large-aperture microphone array," in *ICASSP*. 2016, pp. 440–444, IEEE.
- [2] L. K. Hansen and K. B. Petersen, "Monaural ICA of white noise mixtures is hard," in *Proceedings of ICA'2003 Fourth Int. Symp. on Independent Component Analysis and Blind Signal Separation, Nara Japan, April 4.*, 2003, pp. 815–820.
- [3] Nasser Mohammadiha, Paris Smaragdis, and Arne Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *CoRR*, vol. abs/1709.05362, 2017.
- [4] Jonathan Le Roux, Felix J Weninger, and John R Hershey, "Sparse nmf—half-baked or well done?," *Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA, Tech. Rep., no. TR2015-023*, 2015.
- [5] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.
- [6] 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.
- [7] Y. Luo, Z. Chen, J. R. Hershey, J. Le Roux, and N. Mesgarani, "Deep Clustering and Conventional Networks for Music Separation: Stronger Together," *ArXiv e-prints*, Nov. 2016.
- [8] C. Stephenson, P. Callier, A. Ganesh, and K. Ni, "Monaural speaker separation using source-contrastive estimation," in *2017 IEEE International Workshop on Signal Processing Systems (SiPS)*, Oct 2017, pp. 1–6.
- [9] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2136–2147, 2015.
- [10] Hirokazu Kameoka, *Non-negative Matrix Factorization and Its Variants for Audio Signal Processing*, pp. 23–50, Springer Japan, Tokyo, 2016.
- [11] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, March 2007.
- [12] Mikkel N. Schmidt, "Speech separation using non-negative features and sparse non-negative matrix factorization," in *Computer Speech and Language, 2008, submitted*. [Online]. Available: <http://www.imm.dtu.dk/pubdb/p.php?5377>, 2008.
- [13] Xue Feng, Yaodong Zhang, and James Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1759–1763.
- [14] E. M. Grais and M. D. Plumbley, "Single Channel Audio Source Separation using Convolutional Denoising Autoencoders," *ArXiv e-prints*, Mar. 2017.
- [15] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.
- [16] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey, "Single-channel multi-speaker separation using deep clustering," *CoRR*, <http://arxiv.org/abs/1607.02173>, vol. abs/1607.02173, 2016.
- [17] Zhuo Chen, Yi Luo, and Nima Mesgarani, "Deep attractor network for single-microphone speaker separation," *CoRR*, vol. abs/1611.08930, 2016.
- [18] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," *CoRR*, vol. abs/1607.00325, 2016.
- [19] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.
- [20] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22st ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014.
- [21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 3111–3119. Curran Associates, Inc., 2013.
- [22] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.
- [23] Yuxuan Wang, Arun Narayanan, and DeLiang Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 12, pp. 1849–1858, 2014.
- [24] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections," *ArXiv e-prints*, June 2016.